U.S.A. $3.95 CANADA $4.95
JUNE 1989
VOLUME 8, NUMBER 2

6 Easy-To-Type Programs: 800 XL XE

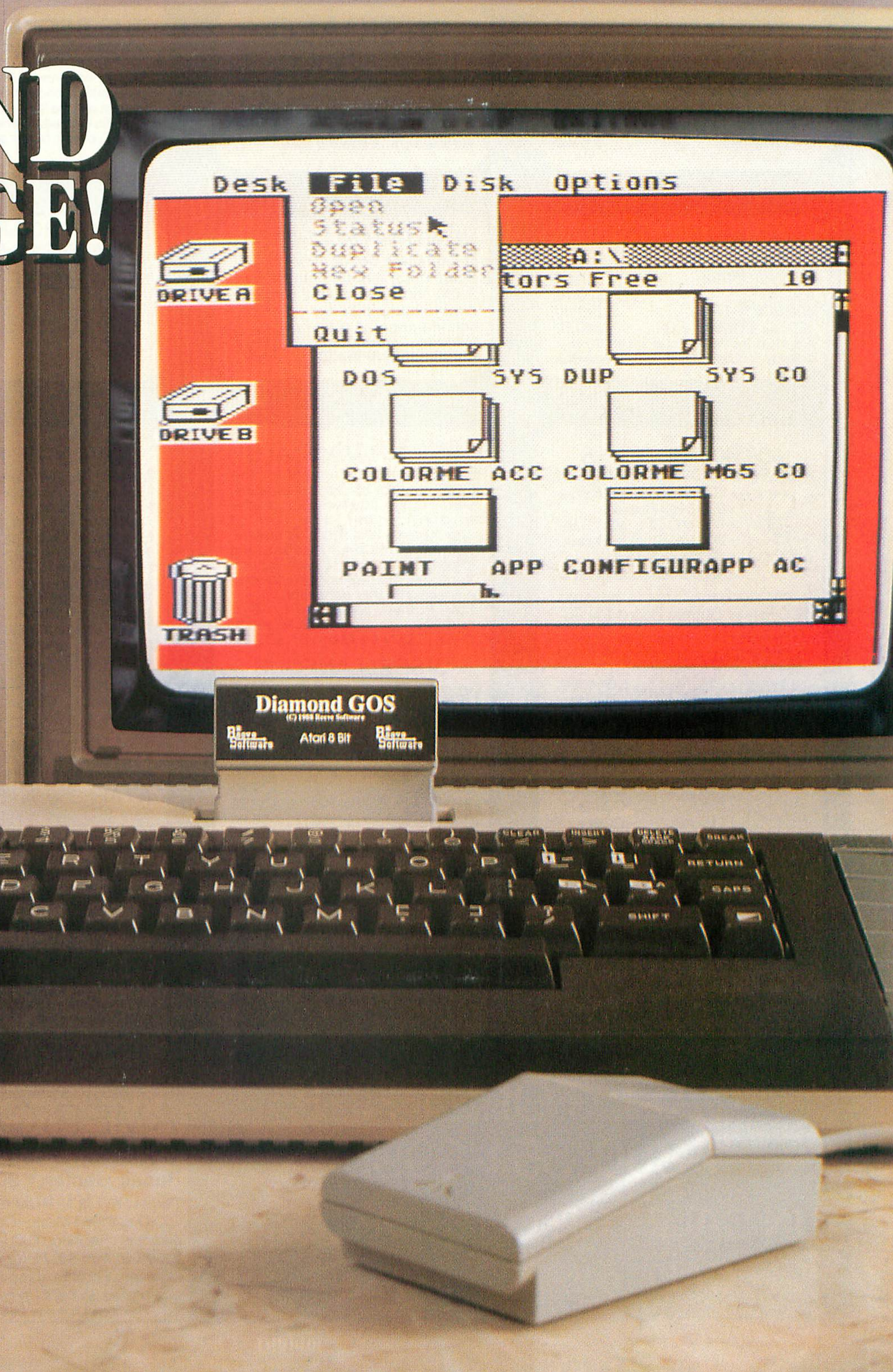
Antic®

The ATARI® Resource

DIAMOND CARTRIDGE!

Turning your 8-bit Atari into ST Jr.

Animated Print Shop Icons



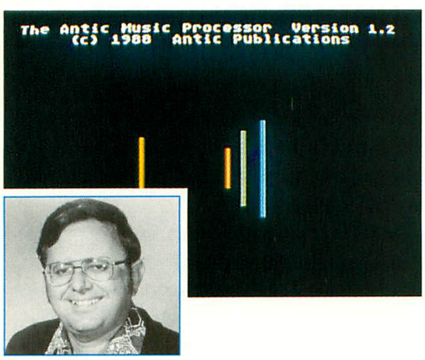
Antic Now with LYRICS! Music Processor See page 23

Antic®

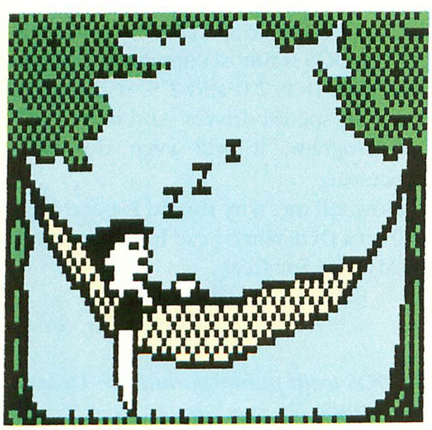
The ATARI® Resource



Diamond Operating System: Page 19



Song Challenge Winners: Page 26



ICON Animation System: Page 15

FEATURES

JUNE 1989, VOL. 8, NO. 2

DEPARTMENTS

SOFTWARE LIBRARY 6 Easy-To-Type 8-Bit Listings

Antic—The Atari Resource (ISSN 0745-2527) is published monthly by Antic Publishing. Editorial offices are located at 544 Second Street, San Francisco, CA 94107. ISSN 0745-2527. Second Class Postage paid at San Francisco, California and additional mailing offices. POSTMASTER: Send address change to **Antic**, The Atari Resource, P.O. Box 1569, Martinez, CA 94553. Subscriptions: One year (12 issues) $28. Canada and Mexico add $8, other foreign add $12. Disk Edition (12 issues with disks) $59.95, all foreign add $25. (California residents add 6½% sales tax for disk subscriptions. Editorial submissions should include text and program listings on disk and paper. Submissions will be returned if stamped, self-addressed mailer is supplied. **Antic** assumes no responsibility for unsolicited editorial material.

Antic

James Capparell
Publisher
John Taggart
Associate Publisher

EDITORIAL
Nat Friedland
Editor
Charles Jackson
Technical and Online Editor
Carolyn Cushman
Assistant Editor
Marta Deike
Editorial Coordinator

ART
Kathleen McKeown
Creative Services Director
Jim Warner
Art Director
Dwight Been
Associate Art Director
Marianne Dresser
Design Production Assistant
Georgia Solkov
Photo Editor and Cover Photography
Julianne Ososke
Production Manager
Kate Murphy
Advertising Production Coordinator

CIRCULATION
Manny Sawit
Director
Amber Lewis
Subscription Coordinator
Dennis Swan
Distribution Coordinator

ADVERTISING
(415) 957-0886
Western Sales Representative
Denny Riley
Eastern Sales Representative
Austin Holian
Sales Coordinator
Diane Van Arsdall

ANTIC PUBLISHING, INC.
James Capparell
President and Chairman of the Board
Donald F. Richard
Richard D. Capparella
Directors
Lee Isgur
Advisor to the Board
John Taggart
Vice President
John Cady
Controller

GENERAL OFFICES
(415) 957-0886
544 Second Street, San Francisco, CA 94107

Credit Card Subscription and Catalog Orders
(800) 234-7001, Visa or MasterCard Only

SUBSCRIPTION CUSTOMER SERVICE
(415) 372-6002
Antic, P.O. Box 1569, Martinez, CA 94553

XETEC DIP-FIX

The March 1989 **Antic** I/O had a fix for running Newsroom page design software with a Xetec Graphix interface on a Panasonic 1080i printer. The number of the DIP switch to be reset was left out. The complete instructions are: Set the Newsroom printer driver to Epson MX-70 and set Xetec *DIP switch 2* to the ON position.—ANTIC ED

LIST MISSED

As a long-time Atari enthusiast and **Antic** contributor, I was interested in your collection of existing products for the 8-bit in the January 1989 issue of Antic. However, you didn't mention my educational chemistry game, **Reaction Time**, distributed by Queue, Inc., 562 Boston Ave., Room S, Bridgeport, CT 96610; (800) 232-2224 or (203) 335-0906. Frankly, I think this is a good and unique product, although the advertised price of $69 may be too high.

In addition, you don't show Dynacomp, Inc. in your list of sources for 8-bit products. Their most recent catalog lists over 70 programs for the Atari. Quite a few are old, but they are inexpensive and *available.*

Karl Wiegers
Rochester, NY

The Dynacomp catalog includes educational, business, finance, statitistics, engineering, science, graphics and utility software—not to mention games. The catalog is available for $2 from Dynacomp, Inc., 178 Phillips Road, Webster, NY 14580. (800) 828-6772, (716) 265-4040.—ANTIC ED

NO DUMPING

IBM PCs and clones have the ability to dump what appears on the screen to a printer, when you press a [PRINT SCREEN] key. Is there some way of doing that with my 800XL? There are so many occasions when I would like to do that without destroying what's on the screen.

Jerry Fraenkel
Franklin Square, NY

Sorry, there's no built-in screen dump for the 800XL or for any 8-bit Atari. We ran an article and program called Kwik Dump *in the March 1985* ***Antic****. But that program only works with graphics file formats—and requires some programming knowledge.*

ANOTHER CONTENDER

I feel compelled to write in response to your article "SpartaDOS X or Atari DOS-XE? Which is Number 1?" I would like to mention a third choice, namely MyDOS 4.5.

MyDOS is density smart, supports RAMdisks of all shapes and sizes, supports double-sided disk drives (like the XF551) and supports hard disk drives and their subdirectories. Best of all, it's free from any of the major commercial information services (GEnie, CompuServe, etc.), users groups, and many BBS's.

Also the source code is included, allowing advanced users to fiddle with the DOS if they like. Unlike SpartaDOS X and DOS-XE, MyDOS is almost completely compatible with DOS 2.0 and 2.5, without the need for special drivers. And with a utility program, it will even do batch processing.

Now tell me, why should I spend even $10 on a DOS when I can have the power of MyDOS for free?

Glenn Garman
East Rochester, NY

MyDOS went public domain in December, 1988. If you don't have access to MyDOS elsewhere, you can still get MyDOS with manual from Newell Industries at 1213 Devonshire, Wylie, TX 75098. (214) 442-6612. A nominal fee is required to cover shipping and handling.—ANTIC ED

TOP DOS?

In his review of SpartaDOS X (March 1989), Matt Ratcliff calls it the best DOS "bar none!" Although I think he writes some great stuff for the Atari, I disagree. I believe TopDOS 1.5+ is *the* ultimate disk operating system for 8-bit Atari computers. Read your own review in the July 1985 issue, *Everything You Wanted to Know About DOS.*

I don't like MS-DOS, and that is why I chose the Atari. I don't need the complex SpartaDOS commands, and I've had it with those who proclaim such complexity to be better.

Dave Bambaloff
Tacoma, WA

Indeed, our 1985 review referred to TOPDOS as "one of the most friendly, full-featured and useful DOS's for the Atari." Unfortunately, as far as we know, TopDOS is no longer commercially available—nor do we know how it performs with the XF551 disk drive. Antic Arcade's Charles Cherry says that TopDOS was his favorite DOS—until SuperDOS came along (which is why the Arcade Catalog now carries SUPERDOS.) —ANTIC ED

SAUCERIAN TROUBLE

In case some of your readers had trouble getting Listing 2 of *Saucerian Shootout* (January 1989) to RUN, there is a simple fix—delete line 1120 and then RUN the program as instructed. This will create a working copy of the game.

John Reiser
Omaha, NE

This letter helped us track down a very obscure and infrequently occurring bug in the program that makes our BASIC loader listings. We've been using that program since 1985, and this is the first time we've had trouble with it. Thanks for letting us know about the problem and sending us the fix!—ANTIC ED

LOVE AFFAIR

What started out as a Christmas gift for my two children has turned into an adventure, a love affair of sorts between me and our Atari XE Game System. I have forsaken sleep, a few mills, and *Wheel of Fortune* to be with my new companion. My lovely wife went so far as to ask if I have named this newcomer to our family.

We bought the XEGS as a substitute for the scarce Nintendo, but quickly tired of the cartridge games (new ones are scarce for this system). Just when we were ready to trade all this for a new go-cart engine, I discovered **Antic** at our local Walden Bookstore. From the December 1988 issue, I ordered a used Indus disk drive from Computer Repeats. It arrived with five utilities and complete instructions. The disk drive gave me a new respect for the Atari XEGS, and the system is just beginning to grow.

The January 1989 **Antic** held an even bigger surprise—one of your Antic Classifieds advertisers was Joe Butner of Oak Hill, WV, about eight miles from me. I got in touch with him and he brought over a copy of his game, ***Fountain of the Gods***. Not only did my family find a great game (best in our collection), but we found a friend with knowledge of our system that he willingly shares.

Jeffery Sanders
Beckley, WV

We're glad to hear that XEGS buyers are discovering the excellent computer hiding inside. Now, if only Atari would do a bit more to spread the word. We could still end up with an 8-bit renaissance!—ANTIC ED

SIDEWAYS PRINT PROBLEMS

Running an expanding business using my Atari 800XL, I found my spreadsheets were getting too big for single-sheet pages, so I recalled a sideways spreadsheet program, *Tapeless Spreadsheet Printer*, from the August 1987 issue. I've typed-in articles before and had no problem running them on my Atari XMM801 printer, but Tapeless was no go. HELP!

Larry Whiting
Surrey, BC, Canada

Tapeless Spreadsheet Printer does not work with ALL Epson-compatible printers. Because the program downloads a sideways character set to the printer, it will only work with a printer that has the built-in RAM to use an alternate character set. The program was not designed for the Atari XMM801, unfortunately.—ANTIC ED

RAMBRANDT IN COLOR

How can I get RAMbrandt to print or dump graphics onto my Star NX-1000 Rainbow printer? Is there a utility program I should be using?

John McCarthy
Brockton, MA

RAMbrandt can only print in black and white, unless you have an Okimate 20 color printer. To get your pictures to print in color, you can save them in Micro Illustrator format from RAMbrandt. Then you can use YEMACYB/4 software ($29.95) from Electronical Software, P.O. Box 8035, Rochester, MI 48063. This color screen dump utility can print your Micro Illustrator formatted pictures in full color. —ANTIC ED

***Antic** welcomes your feedback, but we regret that the large volume of mail makes it impossible for the Editors to reply to everyone. Although we do respond to as much reader correspondence as time permits, our highest priority must be to publish I/O answers to questions that are meaningful to a substantial number of readers.*

Send letters to: Antic I/O Board, 544 Second Street, San Francisco, CA 94107.

LABELMASTER, SMALL BUSINESS SYSTEM

(applications software)
Black Moon Systems
P.O. Box 152
Wind Gap, PA 18091
32K disk

Maintain your mailing lists and print out the labels you want with **LabelMaster** Version 1.6. This powerful tool lets you design labels with up to six lines and a width of 34 characters, and create and edit your data onscreen. **Multi-Column Lister** Version 1.2 makes it possible for you to print your LabelMaster data files in one to four-column formats, on labels or paper. This utility works with Epson (and most compatible) printers. The two are sold as a package, for only $10.

Both LabelMaster and Multi-Column Lister are included in **SBS**, The Small Business System ($30), a group of programs designed to help run a small business. The program modules include Invoicing, Purchasing, Income and Expense Entry, End of Year Inventory, and Reports. Reports can be generated by month, any group of consecutive months, or by year. The Reports Module even includes a Sales and Tax Report and a Profit/Loss Report.

PAGE MARSHAL

(publishing software)
Valar Software
115 Vale Road
Portslade Sussex
BN41 1GE England
26.64 Pounds Sterling (remittance in U.K. funds required.)
XL/XE with minimum 64K, joystick, Epson compatible Dot Matrix printer with interface.

Page Marshal is a suite of programs forming a user-friendly, text-with-graphics Page Processing System. Originally developed as a structured programming chart drafting tool, the system can also produce electronics diagrams, business forms, statistical charts, graphs, tree structure charts and musical scores—not to mention standard uses for desktop publishing, such as advertisements or newsletters. A different character set or font can be used on every line, and a wide range of custom font files is included.

The screen display exactly mimics the printed copy. The program can work with an entire page in RAM, up to 80 columns wide and 80 lines deep. Any DOS 2.5 compatible word processor can produce text files for this system, or text can be entered in Page Marshal's own Typer mode.

PS USER UTILITY DISK

(utility software)
No Frills Software
800 East 23rd Street
Kearney, NE 68847
$31.95, 48K disk

The **PS USERS UTILITY DISK** contains a set of utility programs for use with Broderbund's Print Shop. The icon viewer lets you load four icons at a time and an icon cataloger sets you print specified icons. Font and border catalogers are also included. Other options let you delete, rename or undelete your PS files, or even automatically transfer PS icons from one disk to another. Additional features let you mix PrintShop graphics with Atari fonts, to print custom labels and bookmarks.

EDIT8, THE ENHANCEMENT

(utility software)
Logic One
P.O. Box 18123
Cleveland, OH 44118-0123

Instantly examine and modify your programs with **EDIT8** ($9.95), the one and only RAMdisk sector editor. The editor works with Binary or BASIC programs, text or data, and includes dual-display and a restore feature. According to Logic One, EDIT8 works with all upgrades, under DOS 2.5 or SpartaDOS 3.2.

Also from Logic One, **The Enhancement** ($7.95) lets you use your RAMdisk like a real disk. Working with Atari DOS 2.5 or SpartaDOS, The Enhancement lets DOS know when to recover and when to set up the RAMdisk, making it both automatic and "coldstart safe" — every time.

ATARIWRITER PLUS PATCH

(utility software)
LVAUG
c/o AW+ Patch
P.O. Box 1307
Allentown, PA 18105
$5, SpartaDOS 2.3 or 3.2

Lehigh Valley Atari Users Group (LVAUG) is now offering Craig Gaumer's patch which allows you to use your copy of AtariWriter Plus with SpartaDOS 2.3 or 3.2. With this patch you can run AW+ from a RAMdisk or hard drive, Exit to SpartaDOS and return to AW+ without losing your working copy, read directories of drives 1 through 8 directly from the AW+ menu, and more. Documentation is supplied on disk along with the patch.

*New Products notices are compiled by the **Antic** staff from information provided by the products' manufacturers. Antic welcomes such submissions, but assumes no responsibility for the accuracy of these notices or the performance of the products listed.*

Super LOCATOR

Fast multi-file text finder

By Jeffrey Summers, MD

Ever try to find a specific bit of text when you forgot what file it was in? Save time and frustration by letting Super Locator do the searching for you. This BASIC program works on 8-bit Atari computers with 48K memory and a disk drive.

If you use your Atari 8-bit for word processing, sooner or later you will forget which file contains some vital piece of information. Previously, the only way to find that lost information was to load files one-by-one into your word processor, search for the desired phrase and if it wasn't in that file, move on to the next.

This very time-consuming process can make you wonder why you ever gave up your trusty typewriter to begin with. Now, Super Locator can make finding your text relatively painless. Super Locator will quickly search all the files on your disk for any specified string.

GETTING STARTED

To see how it works, type in Listing 1 and check it with TYPO II. Be sure to SAVE a copy to disk before you RUN it.

If you have trouble typing the special characters in lines 10000-10030, don't type them in. Instead, type Listing 2, check it with TYPO II and SAVE a copy. When you RUN Listing 2, it creates these hard-to-type lines and stores them in a file called LINES.LST.

To merge the two programs, LOAD "D:LOCATOR.BAS" and then ENTER "D:LINES.LST." Remember to SAVE the completed program before you RUN it.

To use Super Locator, RUN the program and insert a disk with several word processing text files on it. Super Locator asks which files it should search, giving you the chance to narrow your search. Simply press [RETURN] to search all the files on the disk in Drive 1.

Entering a directory specifier limits the search and enables you to search the files on a different disk drive. The directory specifier is the same as you would use with DOS. For example, if all of the files you wish to search are on the disk in your second disk drive and end with the extender .TXT, you would enter "D2:*.TXT".

If you could further limit this to files that started with the letter A, you would enter "D2:A*.TXT". To check all of the files on Drive 2, enter "D2:*.*".

Next, you are asked for the string you want to find. This string may have up to 20 characters. The program also asks if the search should be case sen-

sitive, distinguishing between uppercase and lowercase letters. Although you can enter lowercase characters into the string, be careful to use only capital letters when answering other prompts.

End the string with [RETURN]. The program clears the screen and lists the names of the files it is checking as it goes along.

If the program finds your specified string, the string is displayed on the screen along with the text immediately preceeding and following it. This lets you check the context of the phrase you are searching for, to make sure that this is the instance you wanted to find.

For instance, if you were looking for a specific recipe for chocolate chip cookies, on a disk full of recipes, you could search for the string "chocolate". Your results might look like this:

CHECKING D1:DOS.SYS
CHECKING D1:AUTOMATE.TXT
CHECKING D1:LETTER.DOC
CHECKING D1:RECIPE.CPJ
Found in file D1:RECIPE.CPJ
CHARLIE'S FAMOUS CHOCOLATE CAKE THE
VING FROM PANS. CHOCOLATE FROSTING
DIA & KEN'S MOCHA CHOCOLATE CHEESECAKE
ACKAGE SEMI-SWEET CHOCOLATE CHIPS

RECIPE.CPJ looks like a good place to look for that recipe. If you know the *exact* title of the recipe, you could specify "Toll House Cookies," instead. But if your search is *too* narrow, you risk missing your text because of slightly different wording, punctuation or spelling.

Each time the program finds the string, it will ask whether you wish to Continue searching (the program will continue to check that same file for another occurrance), move to the Next file and continue the search, or stop (Exit). When the program has gone through all of the files specified on the disk, the program stops.

PROGRAM LOCATING

Super Locator does not distinguish between word processing files and programs. Thus, you can also use Locator to search a disk for a program that may use a particular prompt. You can look for anything in a quoted string or a BASIC REM statement. You can even search for text within machine language programs. The "context" may be meaningless in these instances, but the program will find the data in any case.

---

*Rochester, New York physician Jeffrey Summers has written numerous useful programs for **Antic**, including* Antic Data-X *(Revision B appeared in the December 1988 issue) and* Job Jar Atari *(January 1989).*

Listing on page 34

AUTORUN Selector

Total control over your settings

By Jason Strautman

Ever known the frustration of trying to get to DOS from a disk with an AUTORUN.SYS file on it? AUTORUN Selector lets you disable the AUTORUN file on booting, with just the press of a button. This short BASIC utility (translated from machine language) works on all 8-bit Atari computers with disk drive.

If you're like me, you've got a lot of disks with AUTORUN.SYS files on them. And if you're like me, sometimes you need the familiar DOS menu for file copying and other housekeeping tasks. So I always had to keep a separate disk with no AUTORUN files, just DOS.SYS and DUP.SYS, for the times I needed the DOS menu. This seemed like a waste of a disk, so I designed a short solution, AUTORUN Selector.

When you append AUTORUN Selector to your favorite AUTORUN.SYS files, the next time you need to go to DOS, just press the [SELECT] key as you boot the disk and AUTORUN Selector will skip over your AUTORUN.SYS file and load DOS.

GETTING STARTED

Type in Listing 1, SELECT.BAS, check it with TYPO II and SAVE a copy before you RUN it. When RUN, SELECT.BAS creates the machine language program, SELECT.EXE, and writes it to your disk. Antic Disk Subscribers will find SELECT.EXE on the monthly disk.

Listing 2, SELECT.M65, is the MAC/65 source code. You do not need to type it in to use AUTORUN Select.

Before AUTORUN Selector will work, you must appended the SELECT.EXE file to the top of the AUTORUN.SYS file you want it to work with. Here's how to do it:

1. Make backup copies of SELECT.EXE and the AUTORUN.SYS file you need to modify. Keep the originals in a safe place—we'll work with the backups.

2. Type DOS to bring up the DOS 2.0 (or DOS 2.5) menu. Make sure SELECT.EXE and your AUTORUN.SYS file are on the same disk.

3. Type the following:

C [RETURN]
AUTORUN.SYS,SELECT.EXE/A [RETURN]

This copies your AUTORUN.SYS file onto the end of SELECT.EXE. The SELECT.EXE file now contains AUTORUN Selector *and* a copy of your AU-

TORUN.SYS file.

4. Delete the AUTORUN.SYS file. You don't need it anymore because SELECT.EXE contains a copy of it.

5. Rename SELECT.EXE to AUTORUN.SYS.

This AUTORUN.SYS file is the finished product. The AUTORUN.SYS file will load and run just as it did before, whenever you boot the disk. If you press the [SELECT] key while booting, AUTORUN Selector will skip over your AUTORUN.SYS file and load DOS instead. Make sure you have the DUP.SYS file on your disk when you do this.

AUTORUN Selector will also work whenever you press [SELECT] and any other console key. For example, XL/XE owners who want to use AUTORUN Selector with a program that doesn't use BASIC can simply press both the [SELECT] and [OPTION] keys while booting.

AUTORUN Selector first checks DOSVEC (10, $0A) for the address to branch to if DOS is selected. Next, it takes the value from CONSOL (53279, $D01F) and determines which console keys are being pressed. If bit 1 of CONSOL is clear, the [SELECT] key is being pressed. This is true even when the [SELECT] key is pressed along with other console keys.

If [SELECT] is *not* pressed, AUTORUN Selector continues to load the rest of your AUTORUN.SYS file.

Otherwise, if [SELECT] is pressed, AUTORUN Selector jumps to the address it took from DOSVEC and loads DUP.SYS which contains the DOS menu.

Jason Strautman lives in San Antonio, Texas and has been programming for the last five years. This is his first appearance in **Antic.**

Listing on page 33

COMING NEXT IN JULY 1989 ANTIC

GLASNOST GAME: FALLING RUSSIAN SQUARES

FlashStick Subliminals

TapeTime Labelers Add-On

SmartRAM 2.5

Customizing the Atari Operating System Device Handlers:

Part 1

By Bob Martin & Martin Mercorelli

This article, the first of a two-part series for experienced BASIC programmers, explains how the Atari operating system handles the various devices attached to your computer and gives you a step-by-step approach to adding a new device handler or modifying an existing one. The BASIC programs work with any Atari 8-bit computer, disk or cassette.

The ability to change your computer's operating system is a very powerful technique. Atari 8-bit owners are in for a treat, because there are several ways to customize your system through software. To illustrate, two programs are included with this article. The first creates a device handler that does nothing. (Believe it or not, that can be useful.) The second modifies the printer handler to print non-printing characters.

The printer handler was written for the Epson RX-80 and the C.Itoh Prowriter, but it should work with any printer capable of dot-graphics printing. Each program illustrates a different aspect of creating device handlers.

What good is adding a device handler to the operating system instead

of having your program perform the same function? To answer this, let's look at how the operating system interacts with the outside world.

CIO ROUTINE

Among the Atari operating system's best features is the way it handles input and output (I/O). All I/O operations are generally performed in the same way, regardless of which peripheral device is accessed—disk drive, keyboard, screen editor or printer. We can simplify I/O because it's normally handled through the Central Input/Output routine (CIO).

In BASIC, I/O is done so naturally that you hardly notice the complexity of what is actually happening. One example is the use of CIO to send data to the printer:

First, open the device through a CIO control block with the command OPEN #3,8,0,"P:". This does three things: 1) it tells the operating system to OPEN CIO control block #3 (IOCB3) for I/O and to prepare for I/O to the printer; 2) the 8 tells the operating system that data will flow from the computer to the printer; and 3) the "P:" tells the computer to send data to the printer. Therefore the OPEN command is for initialization.

Summing up, a device handler simply tells the computer how to talk to a device. The computer needs to know the direction of data flow, which path (or channel) it will use, where to find data, where to put it and how much of it to grab.

DEVICE HANDLERS

If CIO is the computer's I/O interface to the user, then the device handlers are the computer's interface to peripheral devices.

For example, how does the computer send data to the printer? Since the printer *isn't* a disk drive or keyboard, your Atari obviously needs a special routine, which is part of the device handler.

Each device handler has six routines: OPEN, CLOSE, GET, PUT, STATUS, and SPECIAL (See *Figure 1*). In BASIC these machine language routines are controlled through I/O commands. For example, when your program issues the command OPEN #3,8,0,"P:" it is actually using the OPEN part of the printer handler.

Every handler contains six machine language routines and a table containing the address of each routine minus one.

Why the minus one? CIO accesses a function by pushing its address onto the stack, then executing an RTS (ReTurn from Subroutine). The RTS instruction directs the program to the address on the stack *plus* one. To arrive at the correct address, we must compensate by subtracting one from our target address.

Each device handler has six routines.

HATABS

CIO finds the address of the appropriate handler table in the Handler Address Table, HATABS. This table is a 38-byte block of memory occupying locations 794—831 ($031A—$033F).

Each device handler has its own three-byte entry in the Handler Address Table. The first byte is an ASCII character representing the name of the device (K for keyboard, D for disk drive, etc.). The next two bytes hold the address of that device's handler table.

When you issue an OPEN #3,8,0,"P:" command, for example, CIO looks through HATABS for a "P". Then it uses the next two bytes to find the address of the printer's handler table. Once found, CIO searches the printer's handler table for the address of the printer handler's OPEN routine. Finally, CIO executes the printer handler's OPEN routine (See *Figure 2*).

Again, CIO finds the handler in two steps:

- 1) Get the address of the appropriate handler table from HATABS.

- 2) Get the address of the handler routines from the handler table.

WRITING YOUR OWN HANDLER

Now that we know how CIO locates the handler table, here's how to make your own handler:

1) Write the program for the handler. (The handler must have the functions listed in *Figure 1.*)

2) Set up a Handler Table with the address of each function (minus one) in the order given in *Figure 1*. The special function is a jump instruction beginning with a decimal 76, followed by the two-byte address of the special routine.

3) Make an entry in the Handler Address Table for the device.

That's all. Now let's focus on the two examples and how they implement these steps.

NULL HANDLER

Type in Listing 1, NULLHAND.BAS, check it with TYPO II and SAVE a copy before you RUN it. When RUN, NULLHAND.BAS installs the Null Handler, N:, on Page Six. Once installed, you can test the handler by typing:

LIST "N:"

This command LISTs the program to the N: device. If your Atari responds with a "READY" prompt, the handler is properly installed. If you get an error message, however, something went wrong. Check NULLHAND.BAS for typing errors, and try again.

One use of the null handler is to check a disk for a scrambled file (Error 164). One way to check is to use DOS to copy the file to the screen. This, however, is quite time-consuming. Instead, copy the file to the N: device. The Null Handler will

do the job in a jiffy.

To copy a file to the Null Handler, select choice C from the DOS menu and type:

D:*filename*,N:

If the file is read completely without error, then it's intact. All the files on a disk can be checked by typing "D:*.*,N:".

PROGRAM TAKE-APART

Listing 2, NULLHAND.M65, is the MAC/65 assembly language source code for NULLHAND.BAS. You do not need to type it to use the N: handler.

Lines 540-590 contain the N: handler's routines (see step 1, above). Since the N: handler does nothing, all six functions can be handled by the same "do-nothing" routine. This routine does nothing but end with an RTS instruction and return the appropriate error codes to CIO. *All* handler routines must do this.

Ending in an RTS means that CIO pushes the return address on the stack before jumping to the handler. This feature is useful if you're modifying an existing handler because it gives you two entry points.

Error codes are handled the same way. When CIO jumps to the handler, the Y register contains error code 146—"Function not implemented in handler." If the Y register isn't changed, CIO will print "ERROR 146" and return control to you. Since the handler does nothing, the only error message to return is 1, "*No* error has occurred." It should be in both the accumulator and the Y register. So putting a 1 into the A and Y registers is all this handler does before executing the RTS.

Lines 440-520 build a Handler Table for the N: handler (see step 2, above). Since all the functions are the same, there are five NULL-1 values followed by one JMP NULL.

The Handler Table is always that easy to write—although the addresses usually aren't all the same. Just remember to subtract 1 from each of the first five functions.

FIGURE 1

Handler Table Organization & Functions.
OPEN—Initialize the computer for I/O.
CLOSE—Release the resources of the computer for other uses.
GET—Receive one byte from the device. On return, the accumulator holds the byte.
PUT—Put one byte out to the device. The byte is in the accumulator on entry to the put byte routine.
STATUS—Test the readiness of the device.
SPECIAL—Jump to a special routine not covered above.
On entry to the above handler routines, the X register holds the CIO control block number and the Y register contains a 146 (error code for function not implemented in handler). On return to CIO, the accumulator and the Y register contain the appropriate error code.

Lines 220-420 insert the N: device and the address of the N: handler table into the Handler Address Table (See step 3). This is the most complicated part of the program. Lines 220-310 look for a blank spot in HATABS. Lines 350-420 place the ASCII value for N into HATABS, followed by the two-byte address of the N: handler table (low byte, high byte).

Since the letters indicating the devices are separated by a two-byte address, the routine that searches for a blank spot checks every third byte in HATABS.

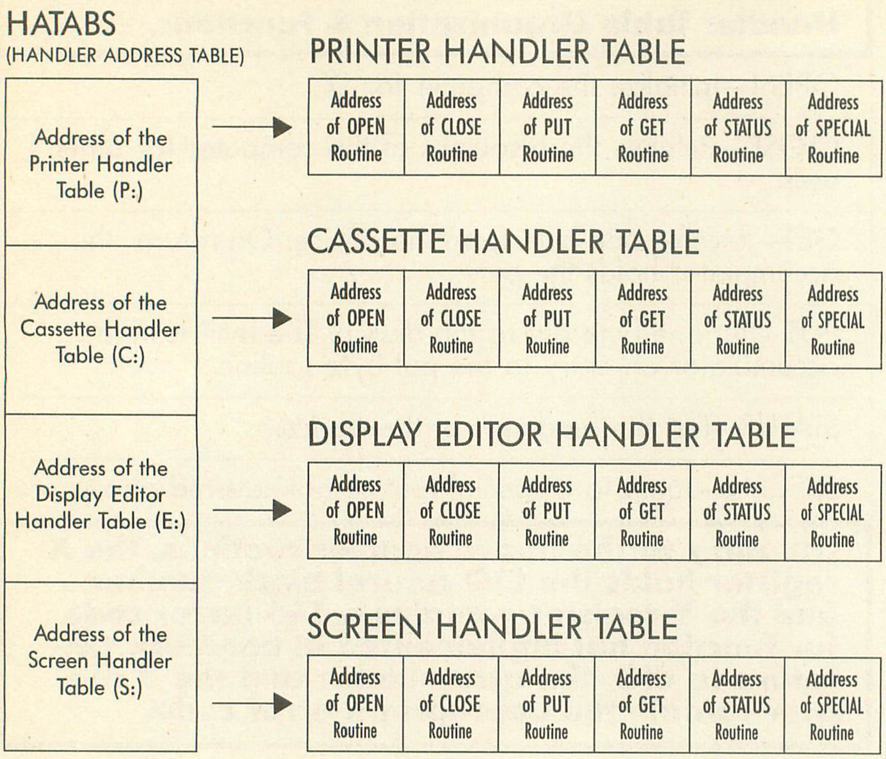
Finally, we place the program's initialization address, INSERT, into INITAD (location 738, $02E2). This forces the program to start as soon as it is loaded into memory. Ending the initialization routine with an RTS returns control to DOS.

USING PRINT HANDLER

The modified printer handler makes it easier to get a hard copy of a file or BASIC program containing non-printing characters (such as the "clear screen" character). Normally, non-printing characters aren't printed. But a special problem arises when the printer recognizes a character as a control character. Control characters are acted upon, not printed. And at the very least, they produce a messy printout. But with the modified printer handler, a neat printout is easily produced.

Type in listing 3, PRHAND.BAS, checking it with TYPO II, and SAVE a copy before you RUN it. When RUN, the program will ask you what type of printer you own. Type [1] if you own an Epson printer, [2] if you own a C.Itoh printer, or [3] for other types of printers. Next, press [RETURN] and the program will create a file called EPSON.EXE (or CITOH.EXE for C.Itoh owners). This is the program which creates and installs the new printer handler. Antic Disk Subscribers will find both versions on the monthly disk.

FIGURE 2



Copy the appropriate .EXE file to a DOS 2.0 or DOS 2.5 disk. Make sure this disk has a DOS.SYS file. Next, rename the .EXE file to AUTORUN.SYS. This disk will automatically install the new printer handler each time you boot your system with it.

Once installed, type POKE 1791,255 to activate the handler. POKE a zero into 1791 to return to normal printer operation. Your Atari will use the modified printer handler until you turn off the computer or press [RESET].

Listing 4, PRHAND.M65, is the MAC/65 source code for the printer handler. You do not need to type it.

TAKE APART

The only difference in our printer handler is the PUT BYTE routine, lines 1030—2570.

Line 290 jumps to the old PUT BYTE routine, and the rest of it just checks for non-printing characters and supplies graphics data.

For the PUT BYTE routine, the byte to be sent is in the accumulator when CIO jumps to the handler. This routine first checks location 1791 ($06FF) to see if the modification is enabled. If so, it then checks to see if the byte to be sent is a non-printing character. For a non-printing character, the control code for eight dot-graphics bytes is sent, followed by the dot-graphics data.

Most printers plot dot graphics the same way. First, the printer is sent a control code enabling dot-graphics mode. On Epsons and compatible printers, the code is 27, 76 ([ESC]-[L]). Then comes the number of bytes the printer should expect. On Epsons and compatibles, a two-byte integer (low byte, high byte) is sent. The complete command for eight-dot graphics mode bytes is 27,76,8,0. On the C.Itoh and compatibles, it's 27,83,48,48,56 (or [ESC]-[S]-[0]-[0]-[8]). Then the eight bytes of graphics data are sent to the printer.

To use character data from the Atari internal character set, we must make it acceptable to the printer. Although your Atari and your printer can define a character in eight bytes, they arrange these bytes differently.

The Atari's character set is organized in horizontal slices starting at the top, and the printer needs vertical slices starting to the left of the character.

To change a screen character data for the printer, each byte for the screen must be sliced one bit at a time, re-formed into eight new bytes and sent to the printer. In other words, bit zero becomes byte one for the printer, bit one becomes byte two, etc.

Lines 210-240 contain the handler table. This table is even simpler than the last one. The new handler table is left blank so that the old table can be copied to it. Notice that the fourth two-byte address (corresponding to the location of the PUT BYTE routine) has a label, making it easy to store the location of the PUT BYTE routine in the new handler table.

Lines 340-800 contain the initialization routine. Lines 340-430 search for the P: handler entry in HATABS. Then lines 460-620 copy the following address to a zero page location and replace the P: handler table address with the address of the new P: handler. The old PUT BYTE address is copied to the jump instruction (line 290) after adding 1. And the new PUT BYTE routine address (minus 1) is placed in the new handler table. Finally, the address of INSERT is stored at $02E2.

NEXT MONTH

Both handlers operate without communicating with the user. The ability to let the user select options is often important for a handler. In Part 2 of the series we'll add I/O to the screen and add everything necessary for a complete handler.

Remember that when writing a device programmer, you only need to worry about half of the I/O conversation—CIO does the rest. ▲

Listing on page 31

2 Column Print Pro

By Earl Haley

HELP files printed on half the pages.

Ever wish you could print the HELP file from the Antic monthly disk in two columns? Two Column Print-Pro prints text files in two columns on most 80-column printers. This handy BASIC program works on all 8-bit Atari computers with at least 32K memory, disk drive and printer.

As a thrifty Atari owner, I don't like wasting half a page of computer paper every time I use my 80-column printer to print 40-column documentation (such as the Antic monthly disk's HELP file). Inspired by the useful *Doc Print-Pro* program in the November 1988 **Antic**, I decided to write a similar program.

Two Column Print-Pro reads a 40-column text file and prints it in two columns on an 80-column printer—without requiring you to rewind the paper. You can print the whole document at once, or have the printer pause after each page. Since the program doesn't use any printer control codes, it should work on just about any 80-column printer.

GETTING STARTED

Type in Listing 1, TWOCOLMN.BAS, check it with TYPO II and SAVE a copy to disk before you RUN it.

If all you ever plan to do with this program is print the HELP file from the Antic monthly disk, the program will tell you all you need to know! Three or four simple keypresses will do it all.

The first screen asks whether you are using standard or legal size paper. Next, choose either continuous feed, or a pause after each page.

To print the HELP file from an Antic disk, put the disk in Drive 1. With the printer ready, choose option [1], ANTIC HELP FILE. The program will read the HELP file and print it in two columns.

PREPARING YOUR OWN TEXT FILES

To print your own document in two columns, load any text file into

```
Most of the programs on this disk are        .BAS   BASIC saved program, use menu
written in Atari BASIC. Some require                RUN or LOAD and RUN.
special hardware or specific operating       .LST   BASIC listed program, ENTER
instructions. Please read the matching              then RUN.
issue of ANTIC Magazine for complete         .EXE   Assembly language 'Load and
documentation.                                      Go' file. Remove BASIC, boot
                                                    DOS, type L then enter the
SUBSCRIPTION Info & Orders:                         filename.
(800) 347-6969.                              .ASM   Assembly language source code
                                                    (Atari Assembler Editor)
BACK ISSUE Orders: (800) 234-7001                   Must be compiled.
(Visa or Mastercard only, please.)           .M65   "SAVEd" MAC/65 assembly
                                                    language source code.
By popular demand, we've included                   Must be compiled.
MORE music files for Antic's AMP
Music Processor (Antic Disk Bonus,           .OBJ   Machine language object file.
Dec. 1988).  This side features                     Same as .EXE but not 'Load and
The Flight Of The Bumble Bee                        Go'. See corresponding article.
(filename "D:BEE").  And we've put           .SYS   Reserved for DOS system files.
a sleighfull of holiday songs on             .TXT   Text file. Usually accessed
the back!                                           by another program.
                                             .ACT   ACTION! language source code.
     (You need the Antic Music                      Must be compiled.
Processor (AMP.EXE) to hear these            .LGO   A Logo program.  Requires the
songs.  It was featured on the                      Logo cartridge.
Dec. 1988 Monthly Disk.)
                                             All programs from ANTIC issues dated
Whenever there is sufficient room,           prior to August 1984 were reproduced
we will be including TYPO II on              from ANTIC's Archive.  We've updated
the monthly disks.  TYPO II is in           them from our HELP columns and
LISTed form, and must be ENTERed             most are compatible with the XL and
before using.  It will not RUN               XE machines. We have worked to insure
from the menu.                               that our programs operate as
                                             published, but we consider all our
The menu that appeared on the screen         software listings to be works in
when you booted the disk will only           progress which we encourage you to
RUN programs with a .BAS extender.           personalize and enhance.
If you try to run a program and the
response is 'cannot run Filename',           Good luck, and ENJOY!
check the file extender with this
list:                                                   -ANTIC ED

                                  - 1 -
                                 - END -
```

your word processor. Change the top margin to 0, the left margin to 0, and the right margin to 38. Then *print the document to disk* as a new file. Be sure to use a different filename if you don't want to write over your present file!

TWOCOLMN.BAS should be able to read and print any 40-column (or less) text file. However, only files with an extender of .TXT will be listed by the program's disk directory, so it's a good idea to add a .TXT extender to the new file.

To preview the printed file, load your new file back into your word processor. Remove any extra carriage returns, and this time *save the file to disk as a regular text file.* (If you print the file to disk again your word processor will probably just put all those unwanted carriage returns back in, to fill out what it thinks are blank lines at the end of pages. As a result, your printout will contain a lot of unwanted blank lines.)

You should now be able to print your file in two columns using TWOCOLMN.BAS. Simply choose option [2], OTHER, when the program asks what file you want to print. At the next prompt, enter the name of your file, and your text will be printed in two-column format. Of course, I can't guarantee that you won't end up with half a sentence all by itself on the last page!

CUSTOMIZING THE PROGRAM

Two Column Print-Pro doesn't contain any machine language routines, so it should be fairly easy to customize. If your printer is capable of elite size type (12 characters per inch), you might insert your own printer control code in the program for a slightly neater printout.

You might also want to adjust the left margin by adding one or two blank spaces in line 290. For example, this line will make your left margin two spaces wider:

```
290 ?#2;"  ";COL1$(S,F);COL2$(S,F)
```

The program removes the XXXs and some of the control codes in the Antic Disk's HELP files. If you would like to have them all print, change line 150 to:

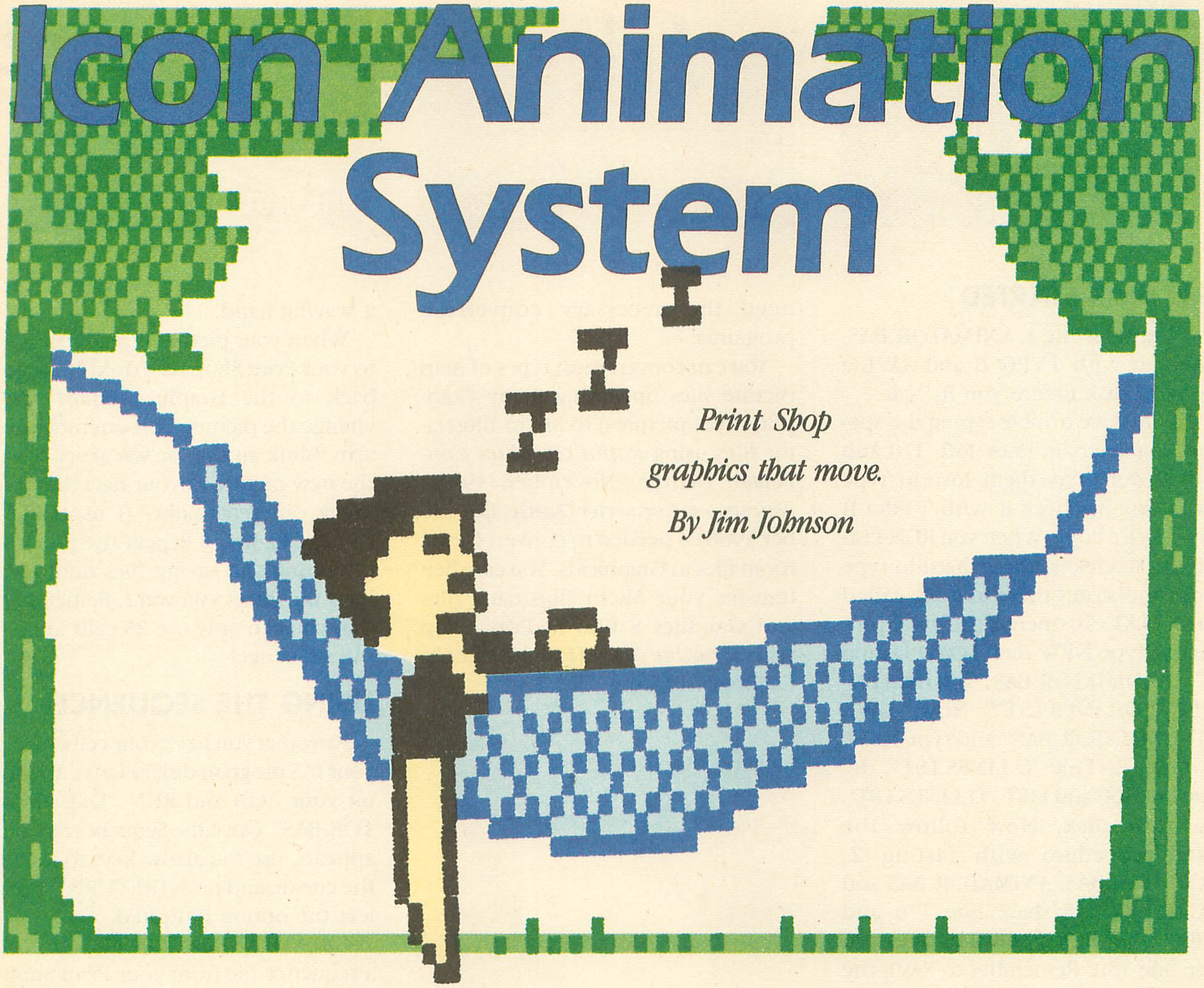
```
150 INPUT #1,R$
```

If you decide to use single sheets of paper (so you can print on both sides of the page), watch out for a possible snag—when I remove the sheet of paper from my printer, the "out of paper" red light comes on. When I put in another sheet of paper, the red light goes out, but I often forget to make sure that my printer is reset to "on". The program just sits there, waiting for me to turn on the printer. If I turn the printer back on, the program resumes where it left off. ▲

Earl Haley lives in Lockport, New York. This is his first appearance in **Antic**.

Listing on page 39

Icon Animation System



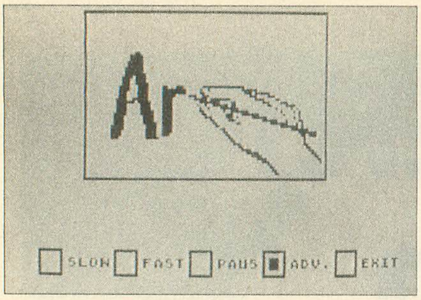
Print Shop graphics that move.

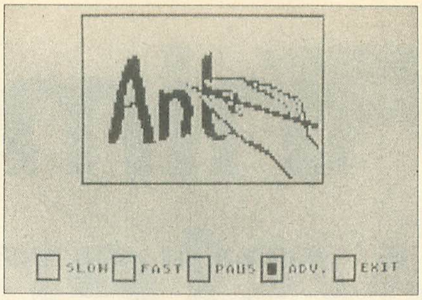
By Jim Johnson

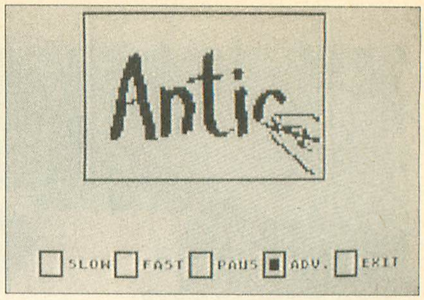
Now you can make easy computer animations using your Print Shop Graphics Editor as a cell animation studio. This two-part BASIC program consists of a Sequencer and a separate Scripter that takes advantage of RAMdisks. It works on 8-bit Ataris with at least 48K memory and a disk drive.

The Icon Animation System makes 8-bit animation easier than ever. You can use it as a tool to learn the basics of cell animation—or simply as an animated billboard. Using the Graphics Editor from Print Shop or Print Shop Companion (Broderbund) as a cell animation studio, you can put together as many as 25 images in a page-flipping animation of up to 100 frames!

The Sequencer module lets you organize your cells in any order, even repeating or skipping frames as desired. Once you have your animations running, the Scripter program lets you link several animations together and run them from disk. You may also run them from a RAMdisk for smoother, faster operation.







GETTING STARTED

Type in Listing 1, ANIMATOR.BAS, check it with TYPO II and SAVE a copy to disk before you RUN it.

If you have trouble typing the special characters in lines 160, 170 and 31000, don't type them. Instead, type in Listing 3, check it with TYPO II and SAVE a copy. When you RUN Listing 3, it creates these hard-to-type lines and stores them in a file called LINES.LST. To merge the two programs, type NEW then LOAD Listing 1, "D:ANIMATOR.BAS" and then ENTER "D:LINES.LST". Now, SAVE "D:ANIMATOR.BAS" and type NEW.

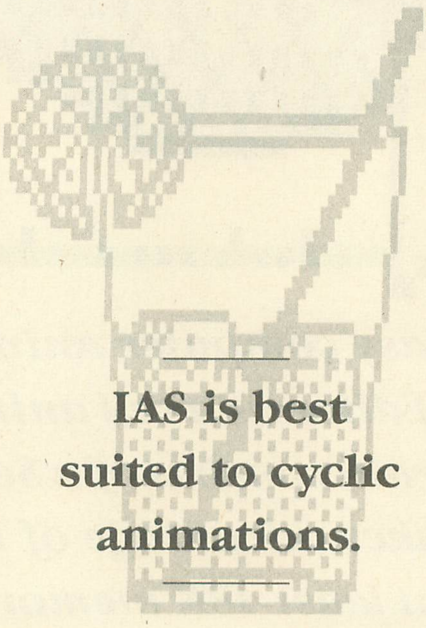
Next, ENTER "D:LINES.LST", delete line 160 and LIST "D:LINES.LST" back to disk. Now follow the same procedure with Listing 2, SCRIPTER.BAS. ANIMATOR.BAS and SCRIPTER.BAS share line 170 and 31000, but SCRIPTER.BAS does not use line 160. Remember to SAVE the completed programs before you RUN them.

MAKING PRINT SHOP CELLS

You will need at least two disks, one formatted from the Print Shop program, another formatted with DOS 2.0 or 2.5. To create the cells or pictures for your animation, start Broderbund's Print Shop or Print Shop Companion and go to the Graphics Editor.

In the Graphics Editor, you can draw an original picture or load any picture from your library of Print Shop icons. Pictures made with other programs can be converted to Print Shop for use with IAS, but you will need the necessary conversion programs.

You can convert most types of Atari picture files (including many ComputerEyes pictures) to Micro Illustrator files using *Rapid Graphics Converter* (**Antic**, November 1985). *Newsroom Converter* (**Antic**, December 1988) is needed to convert Newsroom files to Graphics 8. You can then transfer your Micro Illustrator files and Graphics 8 files to Print Shop with *Graphic Shop* ($19.95, Arcade Catalog, AP156).



IAS is best suited to cyclic animations.

Keep in mind that IAS has a limit of 25 different pictures per sequence—although you can repeat selected pictures within the actual animation, for a total of 100 frames. IAS is best suited to cyclic animation, that is, sequences that repeat themselves, such as a running horse, a juggler, or a waving hand.

When your picture is ready, save it to your Print Shop data disk. Now, go back to the Graphics Editor and change the picture. You can move an arm, blink an eye, or whatever. Save the new picture to your data disk using a different name. (I use CEL1, CEL2, CEL3,. . .) Repeat the process of editing and saving files until you have the effect you want. Remember that you can only use 25 cells in one IAS sequence!

USING THE SEQUENCER

Now that you have your cells, place your IAS program disk in Drive 1, turn on your Atari and RUN "D:ANIMATOR.BAS". Once the Sequencer menu appears, use the arrow keys to move the cursor, and press [RETURN] to select the option indicated.

CREATE ANIMATION FILE creates a sequence file from your Print Shop images. The program asks you for the number of pictures you wish to load, up to a maximum of 25. Then the program will load your files—starting with the *first* icon on the data disk. This way, you do not have to pick and choose each image to be included in your sequence.

Once all the images have been loaded, you go to the Graphics Screen. Here, IAS will flip through your images in the order loaded. The controls below the picture let you adjust the speed of your animation, freeze the action, or advance one by one through the frames.

Use the arrow keys and [RETURN] to control IAS. With the cursor on SLOW or FAST, pressing [RETURN]

will change the animation's speed slightly. Keep pressing [RETURN] until the animation runs at the speed you want. Select PAUSE to freeze the animation, and PAUSE again to continue. ADV will advance your animation one frame and will only work in pause mode. Use EXIT to get back to the main menu.

At any time, you can return to the Graphics Screen from the main menu by choosing VIEW ANIMATION.

The SEQUENCER lets you rearrange the order in which the cells are displayed. Any picture can be used more than once in any given sequence, to produce a more complex animation. While you are limited to 25 different pictures, you can define a sequence of up to 100 frames. Entering a zero as the picture number will insert a pause in the animation. To exit the sequencer, press [RETURN] and you will be returned to the Graphics Screen, where you can view your altered animation.

Once you've finished your sequence, you can save it with the SAVE ANIMATION FILE option. Then you can load your sequence, or any of the samples on this month's Antic Disk, using LOAD ANIMATION FILE. You will be prompted for a filename to load. Press [RETURN] at this prompt to see a directory of the disk in Drive 1.

Animation files consist of the stacked, uncompressed picture and sequence data. When entering a filename be sure to use the device, filename and extension. To load the sample on this month's Antic Disk, put the disk in Drive 1 and at the "FILENAME TO LOAD" prompt type "D:SAMPLE.SEQ".

If you plan to use the Scripter, use .SEQ extenders on your animation filenames. The Scripter looks for a .SEQ extender on sequence files, and .SCR on script files.

While you are limited to 25 different pictures, you can use cells more than once, to define a sequence of up to 100 frames.

USING THE SCRIPTER

With the Scripter program you can go beyond the memory limits of your Atari and create even more complex animations by loading your sequences to a RAMdisk and writing a "script" to run those files in any order. I designed Scripter specifically to run with a RAMdisk.

IAS automatically checks to see if you have a RAMdisk installed. If you don't, the Scripter will load files from Drive 1—but the beauty of the routine is in the speed of a RAMdisk.

When you RUN Scripter, the menu gives you the option to CREATE, LOAD, SAVE and RUN scripts, to exit to the SEQUENCER, or to LOAD FILES TO RAMDISK. If you don't have a RAMdisk, an X will appear in the box opposite this option, and you will not be able to select it.

Whether you're using Drive 1 or a RAMdisk, the sequence files you want to combine all must be on the same disk.

To load files into the RAMdisk you can either SAVE the files to "D8:", or use the Scripter's LOAD FILES TO RAMDISK function. This option will automatically clear the RAMdisk of *all* files (including DUP.SYS and MEM.SAV). It then transfers all sequence files with the extension .SEQ from Drive 1 to the RAMdisk until the RAMdisk is filled.

If you have a small RAMdisk, it may run out of room before all your .SEQ files are copied. To make the best use of your RAMdisk, you should remove any unnecessary .SEQ files from your floppy disk before you begin. (See *Dr. Brilliant's Incredible Atari Brain Transplants* (**Antic**, November 1988) to learn how to expand your RAMdisks.)

If you want to keep the files in the RAMdisk while loading new files, hold down [OPTION] while you select this function, and *keep* it down until the screen displays *LOADING*.

When you choose CREATE SCRIPT, you will be given a listing of the .SEQ files on disk. To create your script, simply pick the sequences you want to play, and then specify how many times that sequence should run before the next sequence loads and plays. Individual sequences can be inserted anywhere in the script and can be used as often as needed.

IAS can be used to set up demos, or as an electronic signboard system. Experienced programmers could extract the animation routines for use in their own programs, including games—and really ambitious programmers could even try adding a sound track. ▲

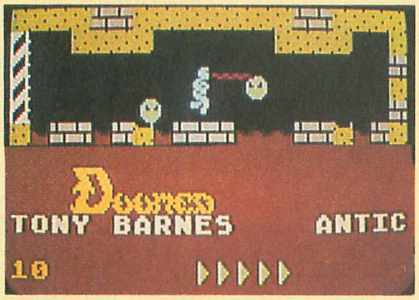
PRINT SHOP, $44.95. Broderbund Software, 17 Paul Drive, San Rafael, CA 94903. (415) 492-3200, (800) 527-6263.

Jim Johnson is a Print Shop addict living in Seaford, Delaware. His program Iconverter *was published in the December, 1987* **Antic**.

Listing on page 35

Doomed

Can YOU survive three minutes in the death pit? **By Tony Barnes**



Doomed is a dodge-and-fight game so challenging that you only need to survive three minutes to win. This BASIC program works on all 8-bit Atari computers having at least 32K memory, with disk or cassette.

They throw you into the cell, and as you hit the cold hard floor they say, "If you can survive three minutes in that death pit you can have your freedom!" Three minutes, you think. Could it be all that bad?

Suddenly, strange beasts come at you from either side. Faster and faster they advance, sapping the very life from you with every pass.

As you concentrate on lasting those three minutes, you notice that one of the beasts disappears, and then another. Then you realize that your mind waves kill these creatures!

GETTING STARTED

To play Doomed, type in Listing 1, DOOMED.BAS, check it with TYPO II and SAVE a copy. If you have trouble typing the special characters in lines 50, 5530-5565, 11000-11010 and 20006-20010, don't type these lines. Instead, type in Listing 2. When you RUN Listing 2, it creates a file called LINES.LST. Merge this file with Listing 1 by typing LOAD "D:DOOMED.BAS and then ENTER "D:LINES.LST". Cassette users: CLOAD Listing 1, then insert the separate cassette used for Listing 2 and ENTER "C:". Remember to SAVE the completed program before you RUN it.

When you RUN Doomed and the main screen appears, press your joystick button to start the game. Pressing the joystick button will also bypass the introductory sequence.

Creatures attack from either side. Any creatures you can't avoid must be shot with your "mind-blaster". To activate it, press the joystick button while moving left or right.

The creatures on the ground are beyond range of your mind-blaster. Your only chance is to jump over them. Move the joystick forward to jump up. As you fall back, you may move left or right, zapping the upper-level creatures and evading the creatures on the floor.

As the game progresses, the ground begins to crumble away. Again, press the joystick forward to jump over the holes. Once you start to fall, move the joystick right or left to scramble onto the stable patches of bricks.

Flashing flags at the bottom of the screen indicate your remaining strength. Once you're down to your last flag, you're doomed to die—and your game is over.

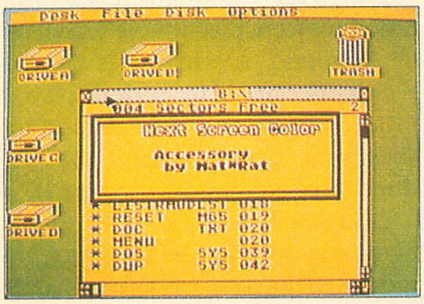
---

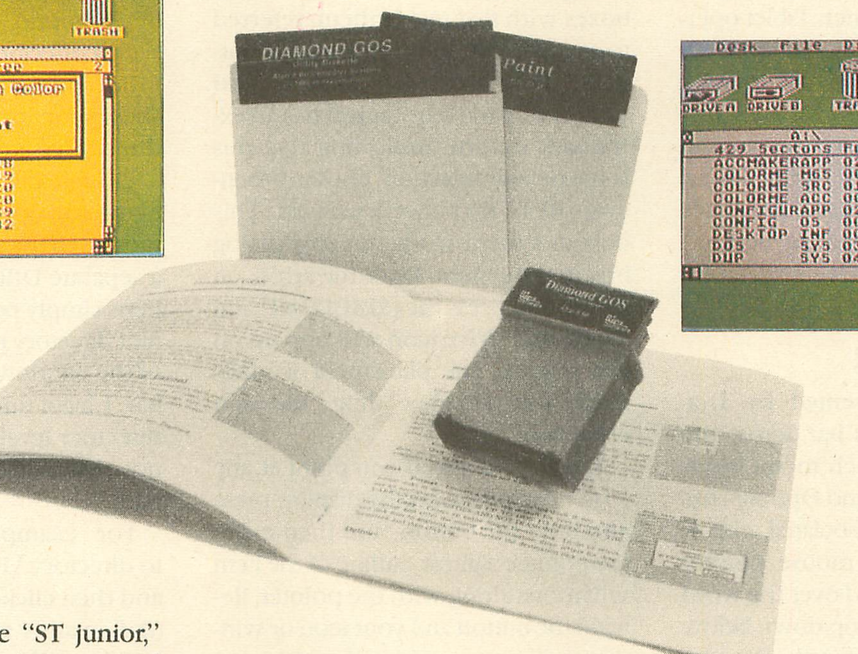
*Tony Barnes is a San Francisco community college freshman. His previous **Antic** games include* Escape From Hell *(June 1988) and* Shutdown *(April 1989).*

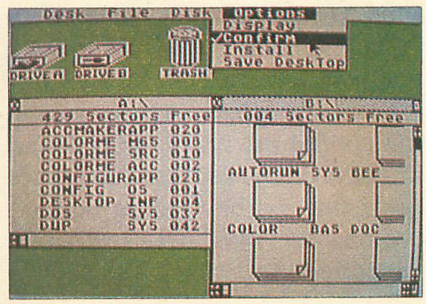
Listing on page 29

DIAMOND OPERATING SYSTEM

Point-and-Click for the Atari 8-bit, Reviewed by Matthew Ratcliff







Touted as the "ST junior," Alan Reeve's long-awaited **Diamond Operating System** is finally here on a bank-switched, piggyback, super cartridge. Diamond gives your 8-bit Atari a high-tech graphic user interface. Typing is replaced with "pointing and clicking." Commands are selected from "drop-down menus" and "pop-up dialog boxes".

First and foremost, Diamond is a replacement for DUP.SYS, the disk utilities package that comes on your Atari DOS disk. The graphic user interface is friendly, almost intuitive. Although Diamond is not exactly a junior ST, it is well executed.

With this cartridge comes a lot of potential. Soon to be delivered in final form, **Diamond Paint** and **Diamond Write** are graphics and desktop publishing-style word processing packages which will run under Diamond.

The advantage of Diamond is that the same graphic user interface can be employed for many applications. This minimizes relearning as you go from one program to the next. Think of all the different programs you use on your 8-bit machine now, each requiring a different command set. Programs that run under Diamond can eliminate all the confusion.

When you first put Diamond to work, you want to configure it for operation under your DOS of choice. I had no problem using the utility programs provided to create a CONFIG.OS file for ICD's SpartaDOS. Dia-

mond even works with the SpartaDOS X cartridge.

It's possible, though rather bizarre, to have MAC/65 piggybacked on top of the Diamond cartridge, on top of the SpartaDOS X cartridge, on top of the R-Time 8 clock cartridge. This is not very practical, however, and I had some problems on my system when running Diamond on top of the SpartaDOS X cartridge (Version 4.19). Diamond does run well from disk-based SpartaDOS 3.2d.

After the Diamond "desktop" is up and running, you need to load and run CONFIGUR.APP from the supplied utility disk. This lets you set up your system for ST mouse, joystick, KoalaPad, or Atari Touch Tablet operation. The default configuration is for an ST mouse in the second joystick port, which worked fine for my setup.

If you do not have a mouse, you can still control the system with the arrow keys on the keyboard, the space bar acting as the "mouse button". Pressing [RETURN] selects "default" menu options.

DESKTOP TOUR

The desktop presented by Diamond places a menu bar across the top of the screen. Each menu heading (Desk, File, Disk, and Options) has a drop-down menu associated with it. For example, if the mouse pointer (black arrow) is moved over the word Options, a box will drop down below it. Underneath, the words Display, Confirm, Install, and Save Desktop will appear.

Moving the arrow over any of these words will highlight it in reverse video text. Press the [SPACEBAR] once (or double-click the mouse, or other external device button) to then activate this option.

For example, highlight Install and press [SPACEBAR]. A dialog box will pop up. This small menu usually comes up centered on the screen, on top of whatever was already there. In this case, the dialog box shows two prompts, Unit Number and Drive Number, each followed by a single blank.

Any time a series of underline characters is displayed within a dialog box, this indicates a field in which some text should be entered. Move the mouse pointer to the Unit Number field and press the [SPACEBAR].

The pointer disappears, replaced by a flashing vertical bar which is the text cursor. Type the number of the drive you wish to install, such as 3, and press [RETURN]. The pointer will reappear, allowing a similar operation for the Drive Letter field. It is only logical to assign drive letter A to drive number 1, B to 2, and so on.

Below these fields are three smaller boxes with text inside them, referred to as "radio buttons" or simply "buttons". They are Install, Remove, and Cancel respectively. Cancel has an extra dark outline, indicating that this is the default selection, chosen whenever [RETURN] only is pressed. This default system works for *any* dialog box in the system. Place the arrow on Install and press the [SPACEBAR] and a new disk drive icon will pop up on the screen. The placement is automatic, but you may "drag" the icon elsewhere.

Drag?, you ask. If you point at any icon or window on the display, press and hold the button, and then move the pointer, a ghost outline of the item will move along with the pointer. Release the button and your icon or window will be transplanted to this new location, if it can fit. The Diamond keeps track of where everything is placed on the screen, and it will not allow icons to overlap.

I found it is best to allow Diamond to automatically place the drive icons. When I tried to place them manually, Diamond was very particular about how close together I set them, which made it very difficult to get my drives (all seven of them) laid out the way I wanted.

ST LOOKALIKE

Diamond provides drop-down menus and dialog boxes in a format very similar to the Atari ST's desktop in low resolution. The icons are oversized, taking up a big chunk of the screen. You won't find the "icon mode" for windows very useful (but then I never did on the ST either). Fortunately, an alternative "text mode" is provided.

At first, double-click response to the mouse was poor. I found myself using the [SPACEBAR] more often than the ST mouse button, until I discovered that the mouse click speed may be adjusted with the CONFIGUR.APP utility, under the Set Mouse Port option.

Whenever you double-click on a disk drive icon, that drive is "opened" by Diamond. A window is created with a directory of that disk displayed inside it. Date and time stamps are shown for SpartaDOS users, and subdirectories are indicated properly.

Double-clicking on a directory name will open up that directory. No more CHDIR command, followed by a separate DIR command is required here, simply point and click. The current directory path is shown at the top of the window, referred to as the title bar. I have found that, when several directory levels deep, Diamond does not always keep track of the current path.

For example, if I went down to directory \FRED\JAMES\BROWN and then clicked on the close button (the small X at the top left of every window), Diamond would indeed go back one directory, as expected. The files listed in the window would be from directory path \FRED\JAMES, but the title bar would only display \FRED. Closing the window and reopening it corrects the title bar.

In the drop-down menu under File, you will find options for opening a file, checking disk status, duplicating disks, creating a new folder, closing a file, or quitting. Selecting Quit will return control to built-in Atari BASIC, or to the cartridge, if one is mounted piggyback on top of the Diamond cartridge.

NO SPEED DEMON

For normal disk maintenance, such as duplicating disks, you may still prefer your old-fashioned Atari DOS or SpartaDOS, since Diamond is much slower in all disk operations by comparison. Duplicating an Atari DOS 2.0S disk with Diamond takes over six minutes (with two drives), when it takes just over two minutes when using Atari's DUP.SYS.

In fact, just about anything you do with Diamond is going to be slower than you are accustomed to. This is the price you pay for all those lovely graphics. Any DOS for your Atari simply manages a 940 byte text display. Diamond must manage nearly 8K of graphics memory, including multiple objects like icons, menus, and windows. It does the job well, but it does take time. The Atari 800XL or 130XE cannot be expected to emulate the ST's desktop at the same speed.

Under the Disk menu you may format a floppy disk or copy files. Files may also be copied by dragging a file from one drive's open window to another. The Options menu allows you to set display preferences. Files may be shown as icons or text. If you choose icons, lots of time will be wasted scrolling around the window. Sort selections for file listings include no sorting (for faster window updates), sort by filename, extender, or file size. I generally sort by name, making files easier to find in a list.

The confirm option has a check mark by it to indicate that Diamond will ask you to confirm any operations that may destroy a file on your disk. Clicking on this entry removes the check mark highlight, indicating that no confirmation requests will be issued.

Install lets you add another disk drive definition, so Diamond will recognize your drives. I find it peculiar that Diamond will allow only seven drives, A-G, to be defined. An eighth drive is not allowed, limiting use with RAMdisks that must be addressed as D8:.

SAMPLE PROGRAM

The file SKELETON.M65 was included in Diamond to assist you in writing your own desktop accessory. The skeleton file was incorrect, and it took a bit of deciphering of RESET.ACC to get it right. I spent several hours studying the source files and came up with COLORME.ACC. This accessory, when activated, will shift the desktop color to the next hue value (same intensity) and display a dialog box. Click the button, and the desktop will return to normal operation in the new color.

I was not successful in developing my own event handling, and did not explore the possibilities of radio buttons and more complicated structures for a dialog box. The source listing for COLORME.M65 is presented below.

```
1000 ; COLORME.M65
1010 ; BY MATTHEW RATCLIFF
1011 ; (c)1989, ANTIC PUBLISHING INC.
1012 ; Use the command:
     ASM D:COLORME.M65,,D:COLORME.EXE
1013 ; To assemble this program.
1020 ;    Note that the FIRST 6 BYTES
1030 ; must be shaved off COLORME.MAC
1040 ; to turn it into an accessory!
1050 ; Use this program:
1060 ; 10 OPEN #1,4,0,"D:COLORME.EXE"
1070 ; 20 OPEN #2,8,0,"D:COLORME.ACC"
1080 ; 30 FOR I=1 TO 6:GET #1,A:NEXT I
1090 ; 40 TRAP 60
1100 ; 50 GET #1,A:PUT #2,1:GOTO 50
1110 ; 60 CLOSE #1:CLOSE #2:? "COLORME
.ACC DONE"
1120 ;
1130      .OPT NO LIST
1140      .OPT OBJ
1160      .INCLUDE #D:LIBRARY.M65
1170      .INCLUDE #D:DMACROS.M65
1190      *=  $2FF0
1200 ;
1210      .BYTE " Color Me " ;10 Byte N
ame
1220      .WORD ENDACC-RUNACC ;Length o
f Accessory
1230      .WORD RUNACC ;Load address
1240      .WORD RUNACC ;Run Address
1250 ;
1260 ;ACCESSORY VARIABLES
1270 COLOREG = 710
1280 ;
1290 RUNACC
1300      DODIALOG  24,60,COLORSOBJ,CO
LORTOUCH
1310      LDA COLOREG
1320      CLC
1330      ADC #$10
1340      STA COLOREG
1350       EVENTDIALOG
1360       RELEASEDIALOG
1370      RTS
1380 COLORS
1390      .BYTE " ",252,4,"Next Screen
Color",13,13
1400      .BYTE 252,1," Accessory",13
1410      .BYTE 252,1,"  by Mat*Rat",13
,0
1420 COLORSOBJ
1430      .BYTE 1
1440      .WORD COLORS,8,8
1450      .BYTE 8,255
1460 COLORTOUCH
1470      .BYTE 0,148,0,136,255
1480 ENDACC
```

MULTI DRIVES

I was able to install both my 10 meg FA-ST hard drive partitions, my XF551 floppy disk, my 1050 disk, and multiple MIO RAMdisks. I was even able to generate a RAMdisk using SpartaDOS's RD.COM, which then resided in the 800XL's internal 256K (RAMbo-XL) memory. Diamond honored this RAMdrive definition and functioned properly with it. Certain memory configurations for Diamond may not allow internal RAMdisks to be used.

When your window placements, drive icon definitions and other preferences are complete, simply select Save Desktop from the Options menu. Diamond will create a file called DESKTOP.INF on the default drive A. The next time your system is booted with Diamond and this disk, the initial display and settings will match those in place when the desktop was saved.

The trash can icon is for deleting files, of course. Simply drag a file from any open window over to the trash can and Diamond will delete it for you.

FEW PROBLEMS

I have not run into too many problems with Diamond. The key to smooth Diamond operation is proper setup with DOSCONFG.BAS and then CONFIGUR.APP.

Diamond usually ran fine with the SpartaDOS X cartridge, but occasionally it would crash. I'm not sure if the problem is with my own highly-customized set-up or not—no one else I talked to had this problem.

Diamond does not return DOS error codes when a disk error occurs. It simply states that "A disk error has occurred". For example, SpartaDOS can access Atari DOS 2.0 and 2.5 disks, with no problems. However, when operating Diamond with SpartaDOS you cannot access disks in Atari DOS format. Diamond simply returns a disk error. Without an associated error code, I have no idea what the problem may be.

I have found that sometimes, when a disk error dialog box pops up, a crash of Diamond soon follows. Under normal operation (working with only one DOS, and disks formatted in that DOS, throughout one session with Diamond), problems are rare.

When two windows are open simultaneously, the bottom window does not get refreshed until it becomes the top (active) window. According to the programmer's reference for Diamond, up to four windows can be opened at once, but the Diamond cartridge allows only two disk drive windows open at any one time. The documentation mentions this, but doesn't explain why. If the cartridge supports four windows, I would expect the desktop to use them as well.

You can "drag" the drive icon to where you want it.

Copying multiple files can be tedious with Diamond. With the Atari ST you can draw a "lasso" box around multiple files, and then drag the whole passel of them over to another window or drive icon for duplication. This "lasso" feature seems to be lacking in Diamond.

The file selector employed in the CONFIGUR.APP program does not allow navigation of subdirectories. Nor does it allow the user to input wild cards to narrow down a set of file selections. Fortunately, this is not built into the Diamond cartridge, but is a function of a separate program file. The program is provided as FILESEL.M65, in MAC/65 source code, on the Diamond utilities disk. Ambitious programmers should be able to perfect it.

Dragging files, icons, and other objects is very basic to the operation of any graphic user interface. You can drag with the joystick, touch tablet, or mouse, but not with the keyboard. The arrow keys, coupled with a console keypress could easily be implemented as a keyboard-controlled drag function, which should have been added for completeness.

The Diamond User's Manual covers the basics of the desktop as briefly as possible. The short manual has no table of contents, nor an index. Screen graphics are provided, which are very helpful for the novice.

The Diamond OS Programmer's Manual is provided for those who wish to write applications or accessories. The manual serves as a fair overview, but is in no way a complete reference guide. However, the source code for CONFIGUR.APP is provided, CONFIGUR.M65. It presents examples of dialog boxes, drop-down menus, a file selector, and more.

With slightly more complete documentation, and final versions of Diamond Paint and Write, Diamond OS is likely to become a very popular operating environment for the 8bit Ataris. The current release is quite workable, and I am eager to see what develops for Diamond. ▲

DIAMOND OPERATING SYSTEM

Reeve Software
29W150 Old Farm Lane
Warrenville, IL 60555
(312) 393-2317

Requires minimum 48K memory.
$29.95 disk
$79.95 cartridge and Programmer's Kit
(Limited introductory special—five units per store—cartridge, Programmer's Kit and Diamond Paint all for $79.95)

Antic Music Processor: Version 2.0

Now you can add lyrics and sound effects, Program by Steven Lashower

Antic Music Processor, the popular December 1988 Super disk bonus, came with a sample song that displayed song lyrics. But there was no apparent way to enter lyrics—author Steven Lashower had plugged in text directly at the programming level. Many Antic Disk users seem to have spent hours staring in amazement at the "Come Sail Away" composition and pondering, "That's great. . .but how can I do it?"

Now, Antic Music Processor, Version 2.0 (AMP) comes with a fully integrated lyric editor to give you that power. Version 2.0 also contains many new enhancements and additional features. Even more than the original version, this new Antic Music Processor is the most powerful music performer available for the 8-bit Atari sound chip. As you'll see in this issue's article about the AMP Song Contest Winners, the program has been used successfully for processing music as complex as Gershwin's *entire* "Rhapsody In Blue."

Your June 1989 Antic Disk—featuring Antic Music Processer, Version 2.0, plus the AMP Song Contest Winners as well as every type-in program from this issue—will be shipped to you within 24 hours after receiving your order. Just phone Toll-Free to the Antic Disk Desk at **(800) 234-7001**. The monthly disk is only $5.95 (plus $2 for shipping and handling) on your Visa or MasterCard. Or mail a $5.95 check (plus $2 shipping and handling) to Antic Disk Desk, 544 Second Street, San Francisco, CA 94107.

The new Note command produces sound effects.

This article explains the new features and commands in Antic Music Processor, Version 2.0. If you *don't* already know how to operate the Antic Music Processor, you'll find full instructions in a large HELP file on this month's Antic Disk. Those on-disk instructions are likely to be necessary for your understanding of the user information that follow. Admittedly, Antic Music Processor is not a program for Atari novices.

By the way, the December 1988 Antic Disk, also still available from the Disk Desk, includes the *AMS File Translator* utility program that converts widely available public domain files from Advanced Music System II into Antic Music Processor format.

Another good reason why you might want to own both versions of Antic Music Processor is that all the new features make Version 2.0 use more memory than its predecessor. This could cause problems loading extra-large song files created with Version 1.2.

You may be able to load massive files and play back most of the song in Version 2.0—but the end of the file might be damaged. Trying to edit over-large files can cause the program to lock up. Instead, load any such files into Version 1.2 and cut them down there. Size limitations will vary, depending on the memory available in your 8-bit Atari computer.

While you are editing your new songs, be sure to *save your files often.* On some rare occasions, Version 2.0 has locked up when the P command is typed in the Note Editor. Nobody has yet been able to get this bug

to repeat consistently, which would be the first requirement for fixing it. Any further information on this will appear, as soon as available, in **Antic** and on CompuServe's ANTIC ONLINE.

USING THE PROGRAM

Copy Antic Music Processor, Version 2.0 to another disk formatted with DOS 2.0 or 2.5. Don't attempt to run it directly from the monthly Antic Disk. Be sure the new disk contains the DOS.SYS file. Use DOS command E to rename AMP2.EXE to AUTORUN.SYS. Turn off your computer and remove all cartridges. Place the disk in drive 1. If you're using an XL or XE, hold down the [OPTION] key while you turn on your computer. Antic Music Processor will load and run automatically.

Two new commands have been added to the Main Menu:

X—Exit to DOS—Press the X key and insert a disk with DUP.SYS to exit to DOS. All music data currently stored in the computer's memory will be erased.

W—Lyric Editor—Press the W key to enter AMP's integrated lyric editor.

L and S—Before, AMP was only able to load and save files to drive one. AMP Version 2.0 allows you to load or save an AMP file from any disk drive. For example, to load "Come Sail Away" from drive 2, you would type:

L D2:COMESAIL.MUS [RETURN].

NEW EDITOR COMMANDS

L—Lyric Advance. Used to display and advance lyrics one syllable at a time.

Nx,y,z—Note command. One of the few complaints about AMP 1.2 was that it lacked the ability to produce sounds other than pure tones. Note is an extremely powerful command that can harness some of the power in the 8-bit Atari's sound generating hardware that until now, has never really been utilized in previous music programs.

AMP's new Note command gives you the ability to produce sound effects and noise. Note is very similar to Atari BASIC's SOUND command. Instead of using four numbers like Atari BASIC, Note uses the following three:

x—Sound pitch (0-255) is exactly the same as the second number in Atari BASICs SOUND command—the lower the number, the higher the frequency of the sound.

y—Sound distortion (0-254).

z—Sound duration (1-126) in clocks. See *Figure 1* for a list of durations and their corresponding clock values.

FIGURE 1

Antic Music Processor Durations

Note			Clock Value of Duration Normal	(.)	(..)	Triplet
Whole Note	(W)	#	192*	xx	xx	128*
Half Note	(H)	#	96	144*	168*	64
Quarter Note	(Q)	#	48	72	84	32
Eigth Note	(E)	#	24	36	42	16
Sixteenth Note	(S)	#	12	18	21	8
Thirty-second Note	(T)	#	6	9	10	4
Sixty-fourth Note	(Z)	#	3	4	xx	2

x—Duration does not exist

*—Since the clock values for these durations are larger than 126, they cannot be used with the Note command.

Let's convert SOUND 0,121,10,8 into a Note command that will be played for 96 clocks, the duration of a half note:

x=121
y=(10 * 16)+8
z=96 (clocks)

The third number in Atari BASIC's sound command (distortion) must be multiplied by 16 and added to the fourth number (volume). The result gives us NOTE 121,168,96, which can be entered into the AMP music editor by typing:

N121,168,96 [RETURN]

LYRIC EDITOR

You can use AMP's integrated Lyric Editor by pressing W from the Main Menu. The Lyric Editor is divided vertically into different sections. In the upper-middle of the screen is a big blue 20 x 11 character window. Up to 11 lines of lyrics can appear in this window at one time.

To the immediate left and right of the window is a small pointer. Pressing [CONTROL] [UP-ARROW] will move the pointer up one line and [CONTROL] [DOWN-ARROW] will move the pointer down one line. The line bracketed by the pointers is the "current lyric line." Then [CONTROL] [LEFT ARROW] and [CONTROL] [RIGHT ARROW] can be used to center the text in the current lyric line.

As you type commands, they will appear in the command line near the bottom of the screen. Error messages will be displayed just below the command line. The status line at the bottom of the screen shows three things:

1. The number of the current lyric line.
2. Entry Mode, either Text or Play.
3. The amount of memory left for entering lyrics.

ENTERING LYRICS

Let's go through the steps to enter the first line of lyrics for the song, "Come Sail Away" — I'M SAILING AWAY. To start out, type C from the Main Menu to clear out any music that might be residing in memory. Now, type W to enter the Lyric Editor.

First of all, we must tell AMP that we're going to enter one line of lyrics. The command for this is L=— followed by up to 20 characters. No more than twenty characters can be displayed on a single line of lyrics. If you type the line below, it will now appear as the top line in the lyric window.

L=I'M SAILING AWAY [RETURN]

Next, we determine where to divide the lyrics up into syllables and breaks: I'M-SAIL-ING-A-WAY. Now we let AMP know about the syllables. Use [CONTROL] [UP-ARROW] to move the pointer back up one line.

Notice that the lyric line is now highlighted right under the window in big, blue Graphics 1 (ANTIC 6) text. Look just below the Graphics 1 line to find the letters which correspond to the syllables and breaks. These letters will be—E, I, N and M. Now type:

S=EINM [RETURN].

The E and M are used because they are spaces separating words, and like syllables, spaces are considered breaks. Press [CONTROL] [RIGHT-ARROW] twice to center the line on the screen. To see how this line will look when played by AMP, type: P [RETURN].

This changes Mode in the status line from Text into Play. Watch the Graphics 1 line and press [RETURN] to simulate a Lyric Advance command. Continue to press [RETURN] until the status line shows Mode: Text.

You can abort play mode by pressing [ESC]. Pressing [ESC] while in text mode will return you to the Main Menu.

LYRIC COMMAND SUMMARY

L= (followed by up to 20 characters) [RETURN]—Enters a line of lyrics into the Lyric Editor.

S= (followed by any sequence of letters from A to T) [RETURN]—Assigns syllables to the highlighted lyric text.

C= (followed by any sequence of letters from A to T) [RETURN]—Removes syllables from the highlighted lyric text.

P [RETURN]—Changes to Play Mode and will perform the highlighted lyric text. Pressing [RETURN] repeatedly simulates the Lyric Advance command.

[CONTROL] [UP-ARROW]—Moves the pointer up one line. [CONTROL] [DOWN-ARROW] moves the pointer down one line

[CONTROL] [LEFT-ARROW] shifts the highlighted text one space to the right. The leftmost character, if any, will now "wrap around" and become the rightmost character. [CONTROL] [RIGHT-ARROW] shifts the highlighted text one space to the left.

[TAB]—Copies the highlighted line of lyrics down to the command line where it can be edited by pressing the [DELETE/BACKSPACE] key.

[CONTROL] [DELETE]—Deletes the highlighted line of lyrics and places it in a one-line buffer. Any previous line stored in the buffer will be lost.

[CONTROL] [INSERT]—Takes the line of lyrics currently in the buffer and inserts it directly above the highlighted line of lyrics.

[CONTROL] [CLEAR]—An alternative to the C= command, clears ALL of the syllables in the highlighted line of lyrics.

[ESC]—While in Play Mode, [ESC] can be used to return to Text Mode. While in Text Mode, hovever, [ESC] is used to return to AMP's Main Menu. A

Steven Lashower studies Computer Science at California State University, Fullerton and works at Disneyland. He would be glad to see your comments and suggestions about Antic Music Processor, if you send them to him in care of **Antic**.

Song Challenge Winners!

''Rhapsody In Blue'' playing on your Atari

*By Carolyn Cushman, **Antic** Assistant Editor*

A*ntic Music Processor,* our December 1988 Super Disk Bonus, proved to be one of **Antic's** most popular programs ever. We received many letters praising the program and suggesting improvements—and dozens of entries for the Antic Song Challenge. The editorial staff had a great time playing all the tunes.

The songs ranged from Vivaldi to Led Zeppelin and Billy Joel. One hardy soul entered Mussorgsky's "Pictures at an Exhibition" in two files for a total of 465 sectors of music. We received ragtime, polka, rock and pop tunes—and even one original symphony. Despite the lack of a lyric editor in our first version of Antic Music Processor, two entries even had lyrics!

Our Grand Winner is G.R. Poole of Virginia Beach, Virginia. His five ragtime tunes, all "Music of Scott Joplin", showed an outstanding sensitivity to dynamics, tempo and sound envelopes. "The Augustan Club", a lively waltz tune, proved a favorite with several editors.

Mr. Poole uses his computer in his work in the beverage wholesale business. He comments:

"I have been an Antic reader since its beginning. Most of the programs never caught my interest. Stephen Lashower's Antic Music Processor is the first program that I can't get enough of. Being a retired Navy Musician who never mastered the keyboard, I have always been frustrated by the fact that the piano and I were not meant to be. Well, thanks to **Antic** I can do what I always wanted to do—play Scott Joplin. Thank you for a superb program."

The winner will receive an Antic Software product from The Arcade Catalog.

RUNNER-UPS

We received so many excellent entries we have eight runner-up winners. All of the runners-up will receive this month's Antic Disk issue.

Unfortunately, many of the best song files we received were very large. There were so many good runner-ups, in fact, that we won't be able to put them all on disk. To fit in as many songs as possible we have even cut some of them to give you a sampling. Full versions of the contest winners will be available for downloading from ANTIC ONLINE this month. After you log onto CompuServe, type GO ANT-2550 and follow the onscreen instructions.

Runner-up Kevin O'Neil of Westminster, Colorado produced the huge "Pictures At An Exhibition" files, definitely pushing the Atari's limits. A programmer by day and a rock keyboard player by night, he demonstrated a real talent for meter and dynamics in translating this complex piece into four voices. Thoughtfully, he also included a thorough description of Mussorgsky's piece, the movements, and the artworks they were based on.

"The piano and I were not meant to be. Well, thanks to AMP I can do what I always wanted to do — play Scott Joplin." Grand Winner G.R. Poole.



G.R. Poole

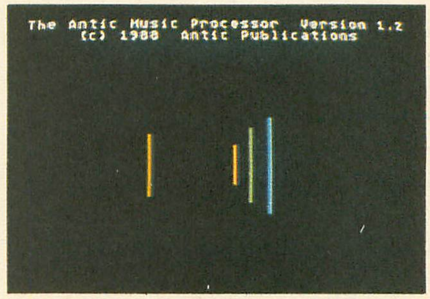
The "Promenade" portion of this piece will be familiar to owners of XE/XL computers—the Audio-Visual Self-Test plays the first measure of this piece. (To see this Self-Test, type BYE from BASIC, then [SELECT] the Audio-Visual test.)

Glenn Saunders of Needham, Mass. sent in six Led Zeppelin tunes (and one from The Monkees). The bouncy Led Zeppelin tune "D'yer Mak'er" was selected as a runner-up. It's definitely got a beat you can dance to!

Lee Vaughan took another runner-up spot with his rendition of a Bach Prelude. We picked this particular piece for its interesting trills and use of contrasting slurs and staccato.

Lyle Davis of Cleveland, Ohio sent in an original composition, "Symphony #1 in A". Considered "interestingly Modern" by one judge, others found it somewhat discordant, but the First Movement, with its heavy bass beat, was considered definitely worth a runner-up position.

Peter Fries of Plano, Texas sent in a long version of Ravel's "Bolero" that deserves extra runner-up credit for including text in the form of a title screen and "program notes" on composer and arranger. This is a much handier system than trying to identify song files with just a file name! At 165



sectors, however, the whole song is too long to include on disk—and cutting it short would remove the gradual build in volume that makes "Bolero" so intense.

Mitch Cohen also produced an impressive, longer runner-up, with some interesting percussion effects—"Roundabout" by Yes.

Cornelius Robinson of St. Rose, Louisiana, entered several classic pieces. We picked his lyrical version of Antonio Vivaldi's "Spring" (from "The Four Seasons") as a runner-up.

Charles Anderson receives a runner-up award for his rendition of "Rhapsody in Blue" by George Gershwin—at 245 sectors, the longest single song file we received.

HONORABLE MENTIONS

Honorable Mentions go to Roger Jensen of West Jordan, Utah, who sent in 13 songs (including "Chopsticks", "Beer Barrel Polka", and "Music Box Dancer"); David Warren of Poway, California ("Black Magic Woman" and "Dance of the Hours"); Nathan Becker of Thousand Oaks, California ("Splish Splash", "Wipe Out" and "Barbara Ann"); Robert Labelle of Ottawa, Ontario, Canada ("Summer of '69", "Stairway to Heaven").

SOFTWARE LIBRARY

TYPING SPECIAL ATARI CHARACTERS

The Atari Special Characters and the keys you must type in order to get them are shown below:

For [CONTROL] key combination, *hold down* [CONTROL] while pressing the next key. For inverse [CONTROL] [A] through [CONTROL] [Z], press the [ ] key—or [ ] on the 400/800—then *release* it before pressing the next key. (Press [ ] or [ ] again to turn off inverse.) For [ESC] key combinations, press [ESC] and then *release* it before pressing the next key.

Carefully study the chart above and pay close attention to differences between lookalike characters such as the slash key's [/] and the [CONTROL] [F] symbol [ ].

NORMAL VIDEO

FOR THIS	TYPE THIS	FOR THIS	TYPE THIS
	CTRL ,		CTRL S
	CTRL A		CTRL T
	CTRL B		CTRL U
	CTRL C		CTRL V
	CTRL D		CTRL W
	CTRL E		CTRL X
	CTRL F		CTRL Y
	CTRL G		CTRL Z
	CTRL H		ESC ESC
	CTRL I		ESC CTRL −
	CTRL J		ESC CTRL =
	CTRL K		ESC CTRL +
	CTRL L		ESC CTRL *
	CTRL M		CTRL .
	CTRL N		CTRL ;
	CTRL O		SHIFT =
	CTRL P		ESC SHIFT CLEAR
	CTRL Q		ESC DELETE
	CTRL R		ESC TAB

INVERSE VIDEO

FOR THIS	TYPE THIS
	ESC SHIFT DELETE
	ESC SHIFT INSERT
	ESC CTRL TAB
	ESC SHIFT TAB
	人 CTRL .
	人 CTRL ;
	人 SHIFT =
	ESC CTRL 2
	ESC CTRL DELETE
	ESC CTRL INSERT

TYPO II AUTOMATIC PROOFREADER

TYPO II automatically proofreads **Antic**'s type-in BASIC listings. Type in the listing below and SAVE a copy to disk or cassette. Now type GOTO 32000. At the prompt, type in a single program line **without the two-letter TYPO II code at the beginning**. Then press [RETURN].

Your line will reappear at the bottom of the screen. If the TYPO II code does not match the code in the magazine, then you've mistyped your line.

To call back a previously typed line, type [*], then the line number, then [RETURN]. When the completed line appears, press [RETURN] again. This is how TYPO II proofreads itself.

To LIST your program, press [BREAK] and type LIST. To return to TYPO II, type GOTO 32000. To remove TYPO II from your program, type LIST "D:FILENAME",0,31999, then [RETURN], then NEW, then ENTER "D:FILENAME", then [RETURN]. Now you can SAVE or LIST your program to disk or cassette.

Don't type the TYPO II Codes!

```
WB 32000 REM TYPO II BY ANDY BARTON
VM 32010 REM VER. 1.0 FOR ANTIC MAGAZINE
HS 32020 CLR :DIM LINE$(120):CLOSE #2:CLO
   SE #3
BN 32030 OPEN #2,4,0,"E":OPEN #3,5,0,"E"
YC 32040 ? "↖":POSITION 11,1:? " TYPO II "

EM 32050 TRAP 32040:POSITION 2,3:? "Type
   in a program line"
HS 32060 POSITION 1,4:? " ":INPUT #2;LINE
   $:IF LINE$="" THEN POSITION 2,4:LIST B
   :GOTO 32060
XH 32070 IF LINE$(1,1)="*" THEN B=VAL(LIN
   E$(2,LEN(LINE$))):POSITION 2,4:LIST B:
   GOTO 32060
TH 32080 POSITION 2,10:? "CONT"
MF 32090 B=VAL(LINE$):POSITION 1,3:? " ";
NY 32100 POKE 842,13:STOP
CN 32110 POKE 842,12
ET 32120 ? "↖":POSITION 11,1:? " TYPO II 
   ":POSITION 2,15:LIST B
CE 32130 C=0:ANS=C
QR 32140 POSITION 2,16:INPUT #3;LINE$:IF
   LINE$="" THEN ? "LINE ";B;" DELETED":G
   OTO 32050
VV 32150 FOR D=1 TO LEN(LINE$):C=C+1:ANS=
   ANS+(C*ASC(LINE$(D,D))):NEXT D
WJ 32160 CODE=INT(ANS/676)
JW 32170 CODE=ANS-(CODE*676)
EH 32180 HCODE=INT(CODE/26)
BH 32190 LCODE=CODE-(HCODE*26)+65
HB 32200 HCODE=HCODE+65
IE 32210 POSITION 0,16:? CHR$(HCODE);CHR$
   (LCODE)
VG 32220 POSITION 2,13:? "If CODE does no
   t match press  RETURN  and edit line a
   bove.":GOTO 32050
```

Can YOU survive three minutes in the death pit?

Doomed

Article on page 18

LISTING 1

Don't type the TYPO II Codes!

```
YV 2 REM DOOMED
UK 4 REM BY TONY BARNES
PF 6 REM (c)1989, ANTIC PUBLISHING INC.
HB 10 GRAPHICS 18:GOSUB 50:GOSUB 20000:GO
   SUB 11000:GOSUB 10000:GOSUB 5500:GOSUB
    40
SX 20 POSITION 3,7:? #6;"&'":POSITION 3,8
   :? #6;"()*+,-"
VU 25 POSITION 0,9:? #6;"tony barnes     a
   ntic";
KH 30 GOTO 8000
EK 40 POKE 708,6:POKE 710,22:POKE 709,10:
   RETURN
TF 50 BRK=USR(ADR("h■■■Mhh■■■■■p■■■■■■■"),
   1):RETURN
IJ 60 FOR T=15 TO 0 STEP -1:SOUND 0,193,1
   0,T:SOUND 1,47,12,T:NEXT T
HQ 65 SC=SC+5:POKE 53278,1:POSITION 0,11:
   ? #6;SC:RETURN
VY 100 IF PEEK(19)>=4 THEN POKE 19,0:LVL=
   LVL+1:IF LVL>8 THEN 1300
AY 105 MN= NOT MN:IF MN THEN 200
NE 110 IF X2<57 OR X2>190 OR Z1=1 THEN 13
   0
YD 115 X2=X2+D2:POKE 53250,X2:SOUND 3,PEE
   K(53770),2,1:RETURN
HE 130 FOR T=200 TO 192 STEP -1:SOUND 0,T
   ,12,2:POKE 706,T:NEXT T
EF 135 POKE 53250,0:POKE 706,200:Z1=0:P2$
   =PM$:I=INT(RND(0)*2)
EY 140 D2=LVL*(I=0)-LVL*(I>0):X2=57*(I=0)
   +190*(I>0):I=INT(RND(0)*3)+1
ZY 145 Y2=31*(I=1)+39*(I=2)+47*(I=3):IF D
   2<0 THEN P2$(Y2)=BR$:RETURN
YQ 150 IF D2>0 THEN P2$(Y2)=BL$:RETURN
PF 200 IF X3<57 OR X3>190 OR Z2=1 THEN 22
   0
ZS 205 X3=X3+D3:POKE 53251,X3:SOUND 2,PEE
   K(53770),2,1:RETURN
JB 210 FOR T=200 TO 192 STEP -1:SOUND 0,T
   ,12,2:POKE 707,T:NEXT T
HT 220 POKE 53251,0:POKE 707,200:Z2=0:P3$
   =PM$:I=INT(RND(0)*2)
GT 225 D3=LVL*(I=0)-LVL*(I>0):X3=57*(I=0)
   +190*(I>0):I=INT(RND(0)*3)+1
EZ 230 Y3=31*(I=1)+39*(I=2)+47*(I=3):IF D
   3<0 THEN P3$(Y3)=BR$:RETURN
BB 235 IF D3>0 THEN P3$(Y3)=BL$:RETURN
DJ 250 I=INT(RND(0)*18)+1:LOCATE I,5,Z:T=
   132*(Z=132 OR Z=162)+162*(Z=161)+35*(Z
   =35)
EB 255 COLOR T:PLOT I,5:RETURN
LK 1000 ST=PEEK(632):GOSUB 100:POKE 53278
   ,1:IF PEEK(53260)>0 THEN GOSUB 1200
ZU 1001 IF LVL>5 THEN GOSUB 250
TH 1002 IF PEEK(644)=0 AND ST<12 THEN 110
   0
TD 1005 X=INT((X1-44)/8):Y=INT((Y1-16)/8)
   :LOCATE X,Y+2,Z:IF Z=132 THEN 1250
GV 1010 IF Z=32 THEN Y1=Y1+1:IF ST>12 THE
   N P0$(Y1)=R1$
CO 1015 IF (ST=10 OR ST=6 OR ST=14) AND Z
   <>32 THEN 1150
VJ 1030 X=X-(ST>8 AND ST<12)+(ST<8):IF X<
   1 OR X>18 THEN 1035
YI 1032 X1=X1-2*(ST>8 AND ST<12)+2*(ST<8)
   :WLK= NOT WLK
MY 1035 IF ST<8 THEN 1060
AO 1037 IF ST>12 THEN 1000
QR 1040 IF WLK=0 THEN P0$(Y1)=R1$
TB 1045 IF WLK=1 THEN P0$(Y1)=R2$
SU 1055 POKE 53248,X1:GOTO 1000
LO 1060 IF WLK=0 THEN P0$(Y1)=L2$
LS 1065 IF WLK=1 THEN P0$(Y1)=L1$
TA 1075 POKE 53248,X1:GOTO 1000
HK 1100 SOUND 0,143,10,3:SOUND 1,147,10,3
   :POKE 53249,X1-17*(ST>8)+9*(ST<8)
ER 1105 GOSUB 100:IF ST>8 THEN P0$(Y1)=R1
   $:P1$(Y1+3)="U■"
TH 1110 IF ST<8 THEN P0$(Y1)=L1$:P1$(Y1+3
   )="■U"
JV 1115 POKE 705,PEEK(20):IF PEEK(53261)=
   4 THEN Z1=1:GOSUB 60
HI 1120 IF PEEK(53261)=8 THEN Z2=1:GOSUB
   60
QK 1125 POKE 53278,1:IF PEEK(53260)>=4 TH
   EN GOSUB 1200
VQ 1130 SOUND 0,0,0,0:SOUND 1,0,0,0:P1$=P
   M$:POKE 53249,255:POKE 53278,1:GOTO 10
   05
KN 1150 FOR Q=0 TO 10:SOUND 0,100,8,10-Q:
   IF X1<57 OR X1>191 THEN 1165
RX 1160 X1=X1-(ST=10)+(ST=6)
HX 1165 Y1=Y1-1:IF ST=14 THEN P0$(Y1)=L2$
DE 1170 IF ST>8 AND ST<12 THEN P0$(Y1)=R1
   $
VB 1175 IF ST<8 THEN P0$(Y1)=L1$
QJ 1180 POKE 53278,1:GOSUB 100:IF PEEK(53
   260)>=4 THEN GOSUB 1200
VZ 1185 POKE 53248,X1:NEXT Q:GOTO 1000
UY 1200 FOR T=15 TO 0 STEP -1:SOUND 1,PEE
   K(53770),10,T:NEXT T
BQ 1205 PWR=PWR-1:IF PWR<-1 THEN 1250
HR 1210 COLOR 32:PLOT 11+PWR,11:RETURN
UZ 1250 POP :FOR I=0 TO 3:SOUND I,0,0,0:N
   EXT I
PA 1255 FOR T=0 TO 15:SOUND 0,PEEK(53770)
   ,8,T:SOUND 1,PEEK(53770),12,T:NEXT T
MY 1260 FOR T=15 TO 0 STEP -1:SOUND 0,PEE
   K(53770),8,T
ZU 1270 SOUND 1,PEEK(53770),12,T:FOR I=70
   4 TO 711:POKE I,T:NEXT I:NEXT T
YZ 1275 FOR T=0 TO 300:NEXT T:P0$=PM$:P1$
   =P0$:P2$=PM$:P3$=PM$:GOTO 8000
ZD 1300 FOR T=140 TO 126 STEP -0.3:SOUND
   0,T,8,2:SOUND 1,T,12,2:POKE 704,T
KX 1305 NEXT T:P0$=PM$:P1$=PM$:P2$=PM$:P3
   $=PM$:FOR T=0 TO 3:SOUND T,0,0,0
AM 1310 POKE 53248+T,0:NEXT T:POKE 559,0:
   POKE 711,146:POSITION 0,0
GP 1315 ? #6;"■■■■################■■■■";
VK 1320 ? #6;"#   ■■■■■■■■■■■■■#■■   ■";
ZT 1325 ? #6;"#                ■■■    ■";
DP 1330 ? #6;"##                      ■";
FZ 1335 ? #6;"###                     ■";
IM 1340 ? #6;"■■■■■■■■■■■■■■■■■■■■■■■■";
OS 1345 P3$(36)=H$:P0$(39)=R1$:POKE 53251
   ,80:POKE 53248,120:POKE 559,46
KI 1350 POKE 707,255:FOR T=0 TO 400:NEXT
   T:TMR=0
KF 1355 IF TMR<60 THEN POSITION 0,11:? #6
   ;"YOU HAVE EARNED "
WQ 1360 IF TMR>60 AND TMR<120 THEN POSITI
   ON 0,11:? #6;"YOUR FREEDOM     "
YU 1365 IF TMR>120 AND TMR<200 THEN POSIT
   ION 0,11:? #6;"FREEDOM???       "
FT 1370 IF TMR>200 AND TMR<250 THEN POSIT
   ION 0,11:? #6;"I WAS INNOCENT   "
SF 1375 IF TMR>250 AND TMR<280 THEN POSIT
   ION 0,11:? #6;"TO BEGIN WITH    "
UH 1380 IF TMR>280 THEN POSITION 0,11:? #
   6;"HA HA HA HA HA"
IL 1385 TMR=TMR+1:POKE 707,248+4*(RND(0)*
   1):IF TMR<310 THEN 1355
RT 1390 GOTO 1260
VC 5500 POKE 559,0:DIM F$(1),F2$((INT(ADR
   (F$)/1024)+1)*1024-ADR(F$)-1),PM$(384)
   ,M$(128)
QC 5505 DIM P0$(128),P1$(128),P2$(128),P3
   $(128)
JT 5510 PM$="♥":PM$(384)="♥":PM$(2)=PM$:M
   $=PM$:P0$=M$:P1$=P0$:P2$=P1$:P3$=P2$
ZE 5515 POKE 54279,ADR(PM$)/256:POKE 5327
   7,3:POKE 623,24:POKE 53257,1
```

continued on next page

```
DO 5520 DIM L1$(18),L2$(18),R1$(18),R2$(1
   8),BL$(12),BR$(12),H$(10)
ZW 5530 L1$="▀▄▀▄▀01r<Lp<▀▀▀▀▀"
HF 5540 L2$="▀▄▀▄▀0Un▀▀>7cc▀▀▀"
CE 5545 R1$="▀p8xP▀6N<2▀<po900▀"
CK 5550 R2$="▀p8xP▀>v▀▀;▀▀▀▀▀▀"
GD 5555 BL$="▀▀<▀▀▀▀▀<▀▀"
GG 5560 BR$="▀▀<▀▀▀▀▀<▀▀"
TW 5565 H$="▀<▀▀▀▀?▀▀":RETURN
MZ 8000 POKE 559,46:GOSUB 40:POSITION 0,0
   :POKE 711,146:POKE 53248,120
GN 8005 ? #6;"▀▀▀▀##############▀▀▀▀";
RI 8010 ? #6;"#  ▀▀▀▀▀▀▀▀▀▀▀▀▀▀   ▀";
CR 8015 ? #6;"#                  ▀";
DN 8020 ? #6;"##                 ▀";
FX 8025 ? #6;"###                ▀";
IK 8030 ? #6;"▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀";
QJ 8035 POKE 53251,80:P3$(36)=H$:P0$(39)=
   R1$:SOUND 0,121,10,4:SOUND 1,162,10,4
SB 8040 SOUND 2,64,12,3:FOR T=0 TO 50:NEX
   T T:POKE 559,46:FOR T=0 TO 50:NEXT T
FX 8045 FOR T=128 TO 140 STEP 0.5:POKE 70
   4,T:SOUND 3,T,8,T-128:NEXT T
FF 8050 FOR T=252 TO 255 STEP 0.5:POKE 70
   7,T:SOUND 3,T,6,T-252:NEXT T
VW 8055 FOR T=4 TO 0 STEP -0.3:SOUND 0,18
   2,10,T:SOUND 1,144,10,T:SOUND 2,64,12,
   T
UG 8060 NEXT T:SOUND 3,0,0,0:TMR=0
TC 8065 POKE 707,248+6*((RND(0)*2)>1):TMR
   =TMR+1
ID 8070 IF TMR>210 THEN POSITION 0,11:? #
   6;"press FIRE to start";:GOTO 8100
BR 8075 IF TMR>60 AND TMR<90 THEN POSITIO
   N 0,11:? #6;"GUILTY AS CHARGED"
YS 8080 IF TMR>90 AND TMR<120 THEN POSITI
   ON 0,11:? #6;"but i am innocent"
VS 8085 IF TMR>120 AND TMR<160 THEN POSIT
   ION 0,11:? #6;"      GUILTY      "
UE 8090 IF TMR>160 THEN POSITION 0,11:? #
   6;"TO THE DOOM CHAMBER"
VT 8092 IF PEEK(644)=0 THEN TMR=210
XH 8095 GOTO 8065
AO 8100 FOR T=140 TO 128 STEP -0.3:POKE 7
   04,T:SOUND 3,T,6,T-128:NEXT T:POKE 704
   ,0
DV 8110 POKE 707,248+6*((RND(0)*2)>1):IF
   PEEK(644) THEN 8110
VL 8115 SC=0:PWR=9:TMR=0:Y1=39:X2=0:X3=0:
   LVL=1:P0$(Y1)=L2$
MT 8120 POKE 704,170:POKE 559,0:POKE 5325
   1,0:POKE 53248,0:POSITION 0,0
IJ 8125 ? #6;"▀▀▀▀##▀▀▀▀▀▀▀▀▀##▀▀▀▀";
NT 8130 ? #6;"▀  ▀▀▀▀        ▀▀▀▀  ▀";
YX 8135 ? #6;"▀                    #";
SM 8140 ? #6;"▀                    ▀";
ZA 8145 ? #6;"▀                    #";
VF 8150 ? #6;"▀▀##▀#▀▀###▀▀▀##▀▀#▀";:POSIT
   ION 0,11:? #6;"0           ":POKE 559,46
YZ 8155 COLOR 192:PLOT 9,11:DRAWTO 19,11:
   FOR T=4 TO 1 STEP -1:COLOR 32
GV 8160 PLOT 0,T:FOR I=15 TO 6 STEP -0.5:
   SOUND 0,237,10,I:SOUND 1,192,12,I
OH 8165 NEXT I:NEXT T:FOR X1=48 TO 64 STE
   P 0.5:SOUND 2,X1,10,2:POKE 53248,X1
ZI 8170 NEXT X1:FOR T=1 TO 4:COLOR 5:PLOT
    0,T:FOR I=6 TO 0 STEP -0.3
OW 8175 SOUND 0,237,10,I:SOUND 1,192,12,I
   :NEXT I:NEXT T:SOUND 2,0,0,0
HI 8180 POKE 711,32:GOTO 1000
BY 10000 RESTORE 11100:READ DNUM
MT 10010 D=PEEK(560)+256*PEEK(561)
OV 10020 FOR I=1 TO DNUM:READ B:POKE D+B,
   PEEK(D+B)+128:NEXT I
AW 10030 B=INT(ADR(DLI$)/256):A=ADR(DLI$)
   -B*256:POKE 512,A:POKE 513,B
TV 10040 POKE 54286,192:POKE 256,Z:RETURN
LI 11000 DIM DLI$(55):DLI$="H▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀
   ▀▀>▀▀Phe▀▀▀▀▀'▀▀Phe▀▀▀▀▀▀▀▀"
JL 11010 DLI$(41)="▀▀▀▀▀▀▀▀▀ ▀▀Phe":RETUR
   N
CD 11100 DATA 3,10,11,12
CA 20000 POKE 559,0:CHR=38912:I=256*224:P
   OKE 756,152
KC 20005 DIM C$(126)
YS 20006 C$(1,63)="▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀
   ▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀?<▀▀?▀▀▀<<▀
   ▀▀▀"
YA 20007 C$(64,126)="▀N▀▀<<▀▀▀▀▀▀▀▀▀▀▀▀▀
   0dSdw>▀▀▀▀ff▀▀▀>cDh▀▀6▀▀n▀g3▀▀ ▀ 0
   8<80 "
ER 20010 MOV=ADR("hh▀▀h▀▀h▀▀h▀▀h▀▀h▀▀▀▀▀
   ▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀")
RU 20015 I=USR(MOV,57344,38912,1024)
RP 20020 FOR X=1 TO LEN(C$) STEP 9
WF 20030 OFFSET=ASC(C$(X,X))
XO 20035 Z=USR(MOV,ADR(C$(X+1)),CHR+OFFSE
   T*8,8):NEXT X
FT 20040 Z=7:RETURN
```

LISTING 2

```
TV 10 REM DOOMED, LISTING 2
VK 20 REM BY TONY BARNES
GX 30 REM (c) 1985,1989 ANTIC PUBLISHING
SA 35 REM (CREATES LINES 50, 5530-5565, 1
   1000-11010 AND 20006-20010 FOR DOOMED.
   BAS)
EV 40 REM (LINES 10-250 MAY BE USED WITH
   OTHER BASIC LOADERS IN THIS ISSUE.
IJ 50 REM CHANGE LINE 70 AS NECESSARY.)
PR 60 DIM FN$(20),TEMP$(20),AR$(93):DPL=P
   EEK(10592):POKE 10592,255
WO 70 FN$="D:LINES.LST":REM THIS IS THE N
   AME OF THE DISK FILE TO BE CREATED
RD 80 ? "Disk or Cassette?";:POKE 764,25
   5
PY 90 IF  NOT (PEEK(764)=18 OR PEEK(764)=
   58) THEN 90
TH 100 IF PEEK(764)=18 THEN FN$="C:"
VB 110 POKE 764,255:GRAPHICS 0:? "    AN
   TIC'S GENERIC BASIC LOADER"
MY 120 ? ,"BY CHARLES JACKSON"
KB 130 POKE 10592,DPL:TRAP 200
PU 140 ? :? :? "Creating ";FN$:? "...plea
   se stand by."
LW 150 RESTORE :READ LN:LM=LN:DIM A$(LN):
   C=1
BQ 160 AR$="":READ AR$
YC 170 FOR X=1 TO LEN(AR$) STEP 3:POKE 75
   2,255
DM 180 LM=LM-1:POSITION 10,10:? "(Countdo
   wn...T-";INT(LM/10);")     "
BK 190 A$(C,C)=CHR$(VAL(AR$(X,X+2))):C=C+
   1:NEXT X:GOTO 160
MM 200 IF PEEK(195)=5 THEN ? :? :? "▀TOO
   MANY DATA LINES!":? "CANNOT CREATE FIL
   E!":END
CM 210 IF C<LN+1 THEN ? :? :? "▀TOO FEW DATA
    LINES!":? "CANNOT CREATE FILE!":END
UU 220 IF FN$="C:" THEN ? :? " Prepare ca
   ssette, press [RETURN]"
AR 230 OPEN #1,8,0,FN$
PV 240 POKE 766,1:? #1;A$;:POKE 766,0
AL 250 CLOSE #1:GRAPHICS 0:? "▀COMPLETED▀
   "
JU 1000 DATA 582
UU 1010 DATA 0530480320660820750610850830
   8204006506808204003410416900013307710 4
   104201000240007169112133016
ZG 1020 DATA 1410142100960340410440490410
   5808206908408508207803215505305305104 8
   032076049036061034000014028
OE 1030 DATA 0300140481081140600761120600
   1424623001201200003415505305305204803 2
   076050036061034000014028030
CF 1040 DATA 0140481241102222222200620550
   9909919519500000341550530530520530320 82
   049036061034000112056120112
VH 1050 DATA 0120540780600500140601121111
   0304804800000341550530530530480320820 50
   036061034000112056120112012
PE 1060 DATA 0621181231230591242361981981
   9519500000341550530530530530320660760 36
   061034000000060126221201255
SS 1070 DATA 2551260600000000341550530530
   5404803206608203606103400000006012618 7
   147255255126060000000034155
FH 1080 DATA 0530530540530320720360610340
   0006012625125225512706303100003405808 2
   069084085082078032155049049
HD 1090 DATA 0480480480320680730770320680
   7607303604005305304105806807607303606 1
   034072173000001230000001141
GR 1100 DATA 0102122010002080071690411410
   2220810406420100120800716903914102220 8
   104064169000141000001169190
XU 1110 DATA 1410341550490490480490480320
   6807607303604005204904106103402220817 3
   020000141024208169032141026
EB 1120 DATA 2081040640340580820690840850
   8207803215505000480480480540320670360 40
   049044054051041061034001251
IX 1130 DATA 2551912555251255223225400212 91
```

```
   |931272552392552542470032232232230002 51
   |25125100000400000000000225243
HV |1140 DATA 2552552550052502422261951341
   |42158191006127063060030014031063127007
   |2402480600600300300015143034
TX |1150 DATA 1550500480480480550320670360
   |40054052044049050054041061034008078014
   |0300600601902551270091350 07
KZ |1160 DATA 0150150301262522480100000481
   |00230115227119062011000218127107195102
   |1021950120000062099242104224
UD |1170 DATA 1230540130002481101951030511
   |23222032000032048056060056048032034155
   |0500480480490480320770790 86
YB |1180 DATA 0610650680820400341041041332
   |41104133240104133213104133212104133239
   |1041332381600001772401452 12
MP |1190 DATA 2302122080022302132302402080
   |02230241198238208234198239016230096034
   |041155
```

Customizing your Atari

Operating System Device Handlers

Article on page 9

Don't type the TYPO II Codes!

LISTING 1

```
OO |100 REM NULL HANDLER INSTALLATION
CE |110 REM BY BOB MARTIN
OI |120 REM (c)1989,ANTIC PUBLISHING INC.
XG |150 RESTORE :TRAP 170:ADDR=1536
YU |160 READ A:POKE ADDR,A:ADDR=ADDR+1:GOT
   |O 160
BZ |170 TRAP 180:XXX=USR(1536)
HI |180 ? :? "N: HANDLER INSTALLED!":END
KH |190 DATA 104,162,0,189,26
MV |200 DATA 3,240,7,232,232,232,208,246,2
   |40,242
XN |210 DATA 169,78,157,26,3,169,31,157,27
   |,3
NO |220 DATA 169,6,157,28,3,96,43,6,43,6
AP |230 DATA 43,6,43,6,43,6,76,44,6,169
FJ |240 DATA 1,168,96,226,2,227,2,0,6
```

LISTING 2

```
0100 ; ADD A NULL HANDLER
0110 ; BY BOB MARTIN
0120 ; (c)1989, ANTIC PUBLISHING INC.
0130 ;
0140 HATABS = $031A
0150 ;
0160        *=  $0600
0170 ;
0180 ;INSERT THE N: DEVICE IN HATABS
0200 ;  LOOK FOR EMPTY SPOT IN HATABS
0210 ;
0220 INSERT
0230        LDX #0
0240 NEXTPLACE
0250        LDA HATABS,X
0260        BEQ ESPOT
0270        INX
0280        INX
0290        INX
0300        BNE NEXTPLACE
0310        BEQ INSERT
0320 ;
0330 ;  FOUND AN EMPYT SPOT
0340 ;
0350 ESPOT
0360        LDA #'N
0370        STA HATABS,X
0380        LDA #NULLT&$FF
0390        STA HATABS+1,X
0400        LDA #NULLT/$0100
0410        STA HATABS+2,X
0420        RTS
0430 ;
0440 ;HANDLER TABLE
0450 ;
0460 NULLT
0470        .WORD NULL-1
0480        .WORD NULL-1
0490        .WORD NULL-1
0500        .WORD NULL-1
0510        .WORD NULL-1
0520        JMP NULL
0530 ;
0540 ;NULL HANDLER
0550 ;
0560 NULL
0570        LDA #1
0580        TAY
0590        RTS
0600 ;
0610 ; RUN INITIALIZATION STEP
0620 ;
0630        *=  $02E2
0640        .WORD INSERT
0650        .END
```

LISTING 3

```
JW |10 REM MODIFIED PRINTER HANDLER
TI |20 REM BOB MARTIN & MARTIN MERCORELLI
AD |30 REM (c)1989, ANTIC PUBLISHING INC.
NG |35 DIM A$(20),PRG$(400),FN$(20)
ES |40 GRAPHICS 0:? :? "     What type of P
   |rinter":? :? :? :? ,"1 - Epson":? ,"2
   |- C. Itoh ProWriter"
UZ |50 ? ,"3 - Other":? :INPUT PR:C=0
JI |60 UPSIDE=0
DM |70 IF PR=3 THEN GOSUB 2000
RA |90 POKE 752,1:? "◙               Working
   |..."
QS |100 READ B:C=C+1:IF B<>-1 THEN PRG$(C,
   |C)=CHR$(B):GOTO 100
IK |200 DATA 255,255,0,6,90,6,0,0,0,0
KC |210 DATA 0,0,0,0,0,0,0,0,0,0
KW |220 DATA 0,0,76,16,6,162,0,189,26,3
JH |230 DATA 201,80,240,5,232,232,232,208,
   |244,189
XQ |240 DATA 27,3,133,205,189,28,3,133,206
   |,169
VF |250 DATA 0,157,27,3,169,6,157,28,3,160
LE |260 DATA 15,177,205,153,0,6,136,16,248
   |,24
RN |270 DATA 173,6,6,105,1,141,17,6,173,7
MW |280 DATA 6,105,0,141,18,6,169,43,141,6
ER |290 DATA 6,169,6,141,7,6,96,226,2,227
SC |300 DATA 2,19,6,19,6,229,6,173,186,6
TD |310 DATA 172,187,6;174,188,6,32;16,6,1
   |52
TE |320 DATA 48,7,142,189,6,140,190,6,96,1
   |04
JL |330 DATA 104,96,141,186,6,140,187,6,14
   |2,188
NX |340 DATA 6,174,255,6,240,21,174,254,6,
   |208
OF |350 DATA 4,201,155,240,12,201,96,240,1
   |2,201
HU |360 DATA 32,144,8,201,123,176,4,32,19,
   |6
SV |370 DATA 96,162,0,142,185,6,189,225,6,
   |32
UM |380 DATA 22,6,174,185,6,232,236,229,6,
   |144
CX |390 DATA 238,173,186,6,32,165,6,133,20
   |4,169
XV |400 DATA 0,133,205,133,206,162,2,24,38
   |,204
WB |410 DATA 38,205,202,16,248,24,169,224,
   |101,205
SF |420 DATA 133,205,162,128,142,185,6,32,
   |191,6
WN |430 DATA 32,22,6,169,0,133,206,78,185,
   |6
WP |440 DATA 174,185,6,208,238,174,189,6,1
   |72,190
```

continued on next page

```
JT 450 DATA 6,152,96,41,127,201,96,176,13
   ,201
NX 460 DATA 32,144,6,56,233,32,56,176,3,2
   4
DJ 470 DATA 105,64,96,0,0,0,0,0,0,169
FD 480 DATA 1,133,203,160,7,138,49,204,24
   0,6
FY 490 DATA 165,203,5,206,133,206,6,203,1
   36,16
WC 500 DATA 240,165,206,44,186,6,16,4,73,
   255,133,206,96
IK 510 DATA 27,76,8,0,4,-1
ZT 1000 IF PR<>2 THEN 1100:REM C.ITOH
RU 1010 PRG$(106,106)=CHR$(231):PRG$(188,
   188)=CHR$(231)
NU 1020 PRG$(281,281)=CHR$(128):PRG$(297,
   297)=CHR$(70)
MV 1030 PRG$(315,319)="50008":PRG$(320,32
   0)=CHR$(6)
AJ 1100 IF PR<>3 THEN 1200:REM OTHER
JD 1110 PRG$(106,106)=CHR$(NUMCOD):PRG$(1
   88,188)=CHR$(NUMCOD)
TG 1120 IF UPSIDE THEN PRG$(281,281)=CHR$
   (128):PRG$(297,297)=CHR$(70)
RM 1130 PRG$(314,313+COUNT)=A$
KD 1200 FN$="D:EPSON.EXE":IF PR=2 THEN FN
   $="D:CITOH.EXE"
GS 1210 IF PR=3 THEN FN$="D:OTHER.EXE"
UI 1220 ? "}":? :? :? "   Creating ";FN$
AK 1230 CLOSE #1:OPEN #1,8,0,FN$:? #1;PRG
   $;:CLOSE #1:POKE 752,0
IA 1240 ? :? :? ,"         DONE!"
HT 1250 IF PR<>3 THEN 1500
QL 1260 ? :? "  NOTE:  If this handler pr
   ints          characters upside down,"
PT 1270 ? "change line 60 to:":?
RF 1280 ? ,"60 UPSIDE=1":?
OF 1290 ? "  Then RUN this program again.
   "
EX 1500 END
XB 2000 ? "}":? :? :? "    Type in the co
   des to put":? "YOUR printer into graph
   ics mode"
OZ 2010 ? :? "  (You can find this inform
   ation":? "in your printer manual)"
CR 2020 ? :? "  Type in the decimal codes
    one at a  time.  End with -1":POKE 75
   2,1
NK 2030 ? :? "EXAMPLE:":? :? "?27":? "?76
   ":? "?8":? "?0":? "?-1":? :? :? "    (P
   ress the SPACEBAR to begin)";
GE 2040 POKE 764,255
AX 2050 IF PEEK(764)<>33 THEN 2050
HP 2060 POKE 764,255:POKE 752,0:COUNT=0:?
    :?
CO 2070 INPUT Q:IF Q<0 THEN 2090
AH 2080 COUNT=COUNT+1:A$(COUNT,COUNT)=CHR
   $(Q):GOTO 2070
ZR 2090 NUMCOD=224+COUNT
AC 2100 RETURN
```

LISTING 4

```
0100 ; MODIFIED PRINTER HANDLER
0102 ; BY BOB MARTIN
0104 ; (c)1989, ANTIC PUBLISHING
0110 ;INITIALIZATION ROUTINE FOR
0120 ; SPECIAL PRINTER HANDLER
0140 HATABS = $031A
0150 ZPAGE1 = $CD
0160 ;
0170      *=  $0600
0190 ;PRINTER HANDLER TABLE
0210 NPTBL
0220      .WORD 0,0,0
0230 NPTBLPC
0240      .WORD 0,0,0,0,0
0260 ;OLD PRINTER PUT BYTE
0280 OLDPNT
0290      JMP OLDPNT
0300 ;
0320 ;INITIALIZE NEW PRINTER HANDLER
0340 INSERT
0350      LDX #0        ;FIND P: HANDLER
0360 PLOOP
0370      LDA HATABS,X
0380      CMP #'P
0390      BEQ PFND
0400      INX
0410      INX
0420      INX
0430      BNE PLOOP
0440 ;
0450 ;MOVE P: HANDLER TABLE TO NEW
0460 ; LOCATION
0480 PFND
0490      LDA HATABS+1,X
0500      STA ZPAGE1
0510      LDA HATABS+2,X
0520      STA ZPAGE1+1
0530      LDA #NPTBL&$FF
0540      STA HATABS+1,X
0550      LDA #NPTBL/$0100
0560      STA HATABS+2,X
0570      LDY #15
0580 PMVLP
0590      LDA (ZPAGE1),Y
0600      STA NPTBL,Y
0610      DEY
0620      BPL PMVLP
0630 ;
0640 ;STORE OLD PUT BYTE ADDRESS THEN
0650 ; CHANGE PUT BYTE ADDRESS TO
0660 ; POINT TO MODIFIED PUT BYTE
0670 ; ROUTINE
0690      CLC
0700      LDA NPTBLPC
0710      ADC #1
0720      STA OLDPNT+1
0730      LDA NPTBLPC+1
0740      ADC #0
0750      STA OLDPNT+2
0760      LDA #[NPUTCH-1]&$FF
0770      STA NPTBLPC
0780      LDA #[NPUTCH-1]/$0100
0790      STA NPTBLPC+1
0800      RTS
0810 ;
0820      *=  $02E2
0830      .WORD INSERT
0850 ;
0860 ;MODIFIED PRINTER HANDLER
0870 ;
0880 ;INTERCEPTS THE PRINTER HANDLER
0890 ; AND SENDS UNPRINTABLE
0900 ; CHARACTERS AS DOT GRAPHICS
0910 ; OTHER CHARACTERS ARE SENT AS
0920 ; NORMAL MODE CHARACTERS
0940 ;
0950 ;POKE 1791 ($6FF) WITH A "0"
0960 ; TO ENABLE.  ANY OTHER VALUE
0970 ; WILL DISABLE THIS OPTION.
0980 ;
0990      *=  INSERT
1000 ;
1010 ;JUMP TO OLD PRINTER PUT BYTE
1020 ;
1030 PNTPUT
1040      LDA HA
1050 PNTPUT2
1060      LDY HY
1070      LDX HX
1080      JSR OLDPNT
1090      TYA
1100      BMI ERROR
1110      STX RX
1120      STY RY
1130      RTS
1140 ERROR
1150      PLA
1160      PLA
1170      RTS
1180 ;
1190 ;MODIFIED PRINTER PUT BYTE
1210 NPUTCH
1220      STA HA        ;CK FOR
1230      STY HY        ;UNPRINTABLE
1240      STX HX        ;CHARACTERS
1250      LDX $06FF
1260      BEQ PNTOK
1270      LDX $06FE
1280      BNE CRGR
1290      CMP #155
1300      BEQ PNTOK
1310 CRGR
1315      CMP #96
1320      BEQ NOPNT
1330      CMP #32
1340      BCC NOPNT
1350      CMP #123
1360      BCS NOPNT
1370 PNTOK
1380      JSR PNTPUT
1390      RTS
1400 ;
1410 ;HANDLER FOR UNPRINTABLE
```

```
1420 ; CHARACTERS
1430 ;
1440 NOPNT
1450      LDX #0
1460 GRLOOP
1470      STX HBIT
1480      LDA GRCODE,X
1490      JSR PNTPUT2
1500      LDX HBIT
1510      INX
1520      CPX CODELN
1530      BCC GRLOOP
1550 ;
1560      LDA HA       ;GET INTERNAL
1570      JSR ASC2SCN  ;CODE # AND
1580      STA $CC      ;CALC ADDR
1590      LDA #0       ;OF FONT
1600      STA ZPAGE1
1610      STA ZPAGE1+1
1620      LDX #2
1630 ALOOP
1635      CLC
1640      ROL $CC
1650      ROL ZPAGE1
1660      DEX
1670      BPL ALOOP
1680      CLC
1690      LDA #$E0
1700      ADC ZPAGE1
1710      STA ZPAGE1
1730 ;
1740      LDX #$80     ;CALC THE 8
1750      STX HBIT     ;BYTES
1760 BLOOP
1770      JSR SLICE
1780      JSR PNTPUT2
1790      LDA #0
1800      STA ZPAGE1+1
1810      LSR HBIT
1820      LDX HBIT
1830      BNE BLOOP
1840      LDX RX
1850      LDY RY
1860      TYA
1870      RTS
1880 ;
1890 ;CONVERT ASCII TO DISPLAY
1900 ; MEMORY VALUE
1910 ;
1920 ASC2SCN
1930      AND #$7F
1940      CMP #96
1950      BCS OUT
1960      CMP #32
1970      BCC GRCHR
1980      SEC
1990      SBC #32
2000      SEC
2010      BCS OUT
2020 GRCHR
2030      CLC
2040      ADC #64
2050 OUT
2060      RTS
2070 HBIT
2075      .BYTE 0
2080 HA   .BYTE 0
2090 HY   .BYTE 0
2100 HX   .BYTE 0
2110 RX   .BYTE 0
2120 RY   .BYTE 0
2130 ;
2140 ;CALCULATES THE VERTICAL-BYTE
2150 ; FROM THE CHARACTER FONT
2160 ;
2170 ;ADDRESS OF CHARACTER
2180 ; IN $CC & ZPAGE1
2190 ;
2200 ;BIT TO BE SENT TO PRINTER
2210 ; IN X REGISTER
2220 ;
2230 ;RESULTING "SLICED" BYTE IS
2240 ; RETURNED IN A AND HELD IN
2250 ; #CE
2260 ;
2270 SLICE
2280      LDA #$01       ;C.ITOH LDA #$80
2290      STA $CB
2300      LDY #$07
2310 LOOP
2320      TXA
2330      AND ($CC),Y
2340      BEQ NOTSET
2350      LDA $CB
2360      ORA ZPAGE1+1
2370      STA ZPAGE1+1
2380 NOTSET
2390      ASL $CB        ;C.ITOH LSR $CB
2400      DEY
2410      BPL LOOP
2420      LDA ZPAGE1+1
2430      BIT HA         ;CK FOR INVID
2440      BPL NOTINV
2450      EOR #$FF
2460      STA ZPAGE1+1
2470 NOTINV
2480      RTS
2490 ;
2500 ;PUT THE CONTROL CODE TO CAUSE
2510 ; YOUR PRINTER TO ACCEPT 8
2520 ; DOT GRAPHICS BYTES HERE
2530 ;
2540 ;C.ITOH CONTROL CODE
2550 ; GRCODE .BYTE 27,"S0008"
2560 ;
2570 GRCODE .BYTE 27,76,8,0
2580 ;
2590 CODELN .BYTE CODELN-GRCODE
2600 ;
2610      .END
```

Total control over your settings

AUTORUN Selector

Article on page 7

LISTING 1

Don't type the TYPO II Codes!

```
VN 10 REM AUTORUN SELECTOR
XM 20 REM BY JASON STRAUTMAN/C. JACKSON
AD 30 REM (c)1989, ANTIC PUBLISHING INC.
TP 40 DIM A$(37)
VX 50 ? "Checking Data...":TRAP 130
YN 60 FOR I=1 TO 37:READ BYTE:TOTAL=TOTAL
   +BYTE*I:A$(I,I)=CHR$(BYTE):NEXT I
PP 70 IF TOTAL<>54579 THEN POKE 195,0:GOT
   O 130
SV 80 TRAP 90
CJ 90 ? "Creating...SELECT.EXE":CLOSE #1
   :OPEN #1,8,0,"D:SELECT.EXE":? #1;A$;
HX 100 CLOSE #1:END
OR 110 DATA 255,255,0,128,24,128,165,10,1
   41,21,128,165,11,141,22,128,173,31,208
   ,41
BU 120 DATA 2,208,3,108,21,128,96,0,0,0,0
   ,226,2,227,2,0,128
FI 130 IF PEEK(195)=0 THEN ? "There is an
    error in the data!!":END
DB 140 IF PEEK(195)=6 THEN ? "Missing Dat
   a!!":END
PU 150 ? "Error ";PEEK(195)
```

LISTING 2

```
0100 ;AUTORUN PICKER
0110 ;BY J.STRAUTMAN/C.JACKSON
0120 ;(c)1989, ANTIC PUBLISHING INC.
0130 CONSOL = $D01F    ;Console keys
0140 DOSVEC = $0A      ;Load DUP.SYS
0150      *=  $8000
0160 START
0170      LDA DOSVEC   ;Grab current
```

continued on next page

```
0180     STA VECTOR  ;DOS vector
0190     LDA DOSVEC+1
0200     STA VECTOR+1
0210     LDA CONSOL  ;CONSOL key
0220     AND #$02    ;Is it SELECT?
0230     BNE CONT
0240     JMP (VECTOR) ;Load DUP.SYS
0250 CONT RTS              ;Load AUTORUN
0260 VECTOR
0270         .WORD 0,0
0280         *=  $02E2
0290         .WORD START
0300         .END
```

Fast multi-file text finder

Locator

Article on page 5

LISTING 1

Don't type the TYPO II Codes!

```
BZ 10 REM LOCATOR
JR 20 REM JEFFREY A. SUMMERS, MD
AD 30 REM (c)1989, ANTIC PUBLISHING INC.
CH 100 DIM FN$(20),FN1$(20),MASK$(20),SEA
   RCH$(50),WINSTR$(143),CASE$(1),BUFR$(1
   0000)
KI 109 GOSUB 10000:FOUND=0
RO 110 CLOSE #1:OPEN #1,12,0,"E:":POKE 71
   0,18
EM 120 DLIST=PEEK(560)+256*PEEK(561):POKE
    DLIST+3,71:POKE DLIST+6,6:? "     LOCA
   TOR                by J Summers"
WS 130 POSITION 7,7:? "Files to search: "
   ;:INPUT #1,MASK$
MA 140 POSITION 7,9:? "String to search f
   or: ";:INPUT #1,SEARCH$
GW 150 POSITION 7,12:? "Case sensitive (Y
   /N): ";:INPUT #1,CASE$:IF CASE$<>"Y" A
   ND CASE$<>"N" THEN 150
RN 160 IF MASK$="" THEN MASK$="D:*.*"
UA 165 PTR=USR(ADR(WINSTR$),ADR(MASK$),LE
   N(MASK$),ADR(":"),1,2)
XJ 170 SLEN=LEN(SEARCH$):WINSTR=ADR(WINST
   R$):BUFR=ADR(BUFR$):BUFR$(10000)=" "
AT 180 SEARCH=ADR(SEARCH$)
ES 200 TRAP 1000:FPTR=1:GRAPHICS 0:POKE 7
   10,18
RN 210 CLOSE #2:OPEN #2,7,0,MASK$
BU 220 TRAP 600:FOR I=1 TO FPTR:INPUT #2,
   FN$:NEXT I:FN1$=MASK$(1,PTR):FN1$(PTR+
   1)=FN$(3,10):FPTR=FPTR+1:CLOSE #2
KE 230 FOR I=LEN(FN1$) TO 1 STEP -1:IF FN
   1$(I,I)<>" " THEN POP :GOTO 240
GN 235 NEXT I
SE 240 FN1$(I+1)=".":FN1$(I+2)=FN$(11,13)
FJ 250 CLOSE #3:OPEN #3,4,0,FN1$
RR 260 ? "CHECKING ";FN1$
PO 300 FLEN=USR(INP,3,BUFR,10000)
CD 310 IF CASE$<>"Y" THEN D=USR(MAKEUP,BU
   FR,FLEN)
CT 320 P=USR(WINSTR,BUFR,FLEN,SEARCH,SLEN
   ,1)
OZ 330 IF P<>0 THEN 400
PL 340 IF FLEN<>10000 THEN 500
DR 350 BUFR$(1)=BUFR$(9900):BUFR$(10000)=
   " "
WP 360 FLEN=USR(INP,3,BUFR+100,9900)+100
NB 370 GOTO 310
SO 400 BEGIN=P-18:IF BEGIN<1 THEN BEGIN=1
XC 410 ENN=P+19:IF ENN>FLEN THEN ENN=FLEN
WW 415 ? "Found in file ";FN1$;"        ":
   FOR I=BEGIN TO ENN:? CHR$(27);BUFR$(I,
   I);:NEXT I:FOUND=1
ZN 420 POKE 752,1:? :? "Next file Exit Co
   ntinue":POKE 752,0
UT 430 CLOSE #5:OPEN #5,4,0,"K:":GET #5,X
   :CLOSE #5:IF X=78 THEN 500
UA 440 IF X=69 THEN 600
PR 450 IF X<>67 THEN 430
SX 460 P=USR(WINSTR,BUFR,FLEN,SEARCH,SLEN
   ,P+1)
NY 470 GOTO 330
ZH 500 CLOSE #3:GOTO 210
OA 600 IF  NOT FOUND THEN ? :? "Not Found
   ."
BC 610 ? "Done.":END
WU 1000 ? :? "Illegal directory name":GOT
   O 2000
JP 2000 FOR I=1 TO 700:NEXT I:CLOSE #1:CL
   OSE #2:GOTO 110
FK 10000 INP=ADR("hhh[illegible]
   [illegible]")
TW 10010 WINSTR$(1,71)="hh[illegible]
   [illegible]
   [illegible]"
XO 10020 WINSTR$(72,143)="[illegible]
   [illegible]
   [illegible]"
AX 10030 MAKEUP=ADR("hh[illegible]
   [illegible]
   8[illegible]")
DN 10040 RETURN
```

LISTING 2

```
RY 10 REM LOCATOR, LISTING 2
JR 20 REM JEFFREY A. SUMMERS, MD
GD 30 REM (c) 1985,1988 ANTIC PUBLISHING
SK 35 REM (CREATES LINES 10000-10030 FOR
   LISTING 1, LOCATE.BAS)
EV 40 REM (LINES 10-250 MAY BE USED WITH
   OTHER BASIC LOADERS IN THIS ISSUE.
IJ 50 REM CHANGE LINE 70 AS NECESSARY.)
PR 60 DIM FN$(20),TEMP$(20),AR$(93):DPL=P
   EEK(10592):POKE 10592,255
WU 70 FN$="D:LINES.LST":REM THIS IS THE N
   AME OF THE DISK FILE TO BE CREATED
RD 80 ? "Disk or Cassette?";:POKE 764,25
   5
PY 90 IF  NOT (PEEK(764)=18 OR PEEK(764)=
   58) THEN 90
TH 100 IF PEEK(764)=18 THEN FN$="C:"
VB 110 POKE 764,255:GRAPHICS 0:? "     AN
   TIC'S GENERIC BASIC LOADER"
MY 120 ? ,"BY CHARLES JACKSON"
KB 130 POKE 10592,DPL:TRAP 200
PU 140 ? :? :? "Creating ";FN$:? "...plea
   se stand by."
LW 150 RESTORE :READ LN:LM=LN:DIM A$(LN):
   C=1
BQ 160 AR$="":READ AR$
YC 170 FOR X=1 TO LEN(AR$) STEP 3:POKE 75
   2,255
DM 180 LM=LM-1:POSITION 10,10:? "(Countdo
   wn...T-";INT(LM/10);")     "
BK 190 A$(C,C)=CHR$(VAL(AR$(X,X+2))):C=C+
   1:NEXT X:GOTO 160
MM 200 IF PEEK(195)=5 THEN ? :? :? "TOO
   MANY DATA LINES!":? "CANNOT CREATE FIL
   E!":END
CM 210 IF C<LN+1 THEN ? :? "TOO FEW DATA
    LINES!":? "CANNOT CREATE FILE!":END
UQ 220 IF FN$="C:" THEN ? :? " Prepare ca
   ssette, press [RETURN]"
AR 230 OPEN #1,8,0,FN$
PV 240 POKE 766,1:? #1;A$;:POKE 766,0
AL 250 CLOSE #1:GRAPHICS 0:? " COMPLETED 
   "
KD 1000 DATA 339
EU 1010 DATA 0490480480480480320730780800
   61065068082040034104104104010010010010
   170169007157066003104157069
KT 1020 DATA 0031041570680031041570730031
   04157072003032086228189072003133212189
   073003133213096034041155049
QS 1030 DATA 0480480490480320870730780830
   84082036040049044055049041061034104104
   133215104133214104133225104
FP 1040 DATA 1332241041332171041332161041
   04133226104133231104133230056165230233
   001133230165231233000133231
QT 1050 DATA 0241652301012141332321652311
```

```
   01215133233024165214101224133228165215
   101225133229160000177034155
UF 1060 DATA 0490480480500480320870730780
   83084082036040055050044049052051041061
   034232209216208032200196226
CH 1070 DATA 2082451652320562292141332121
   65233229215133213165212024105001133212
   165213105000133213096165232
QS 1080 DATA 1972282080131652331972292080
   07169000133212133213096165232024105001
   133232165233105000133233024
GS 1090 DATA 1441810341550490480480510480
   320770650750690850800061065068082040034
   104104133213104133212104133
DE 1100 DATA 2151041332141652120241012141
   33216165213101215133217160000177212201
   097016024165212024105001133
NB 1110 DATA 2121652131050001332131972172
   08233165212197216208227096056233032145
   212024144224034041155
```

Print Shop Graphics that move

Icon Animation System

Article on page 15

LISTING 1

Don't type the TYPO II Codes!

```
LZ 100 REM ICON ANIMATION SYSTEM,
        LISTING 1, THE ANIMATOR
BO 103 REM BY JIM JOHNSON
HM 106 REM (c)1989 ANTIC PUBLISHING INC.
VQ 110 C0=0:C1=1:C2=2:C3=3:C4=4:C5=5:C6=6
   :C7=7:C8=8:C9=9:C20=20:MAX=C20+C5
OH 120 MODE=C6:OFF=40000:WAIT=10
EN 130 DIM FILENAME$(20),KEY$(C3)
AU 140 DIM B$(128),G$(616),G2$(616*MAX)
DJ 150 DIM SEQ$(101),SCRIPT$(10*10)
QA 160 SRADR=ADR("hh▮h▮h▮▮▮h▮▮▮h▮▮▮hh▮▮▮h
   h▮▮R▮▮▮ S▮J▮▮▮I▮P▮▮▮▮P▮▮▮▮U▮▮▮▮T▮")
YX 170 PICT=ADR("hh▮Lh▮Kh▮Nh▮Mhh▮0hh▮hh▮P
   S▮▮1M▮K▮▮▮▮P▮▮▮K▮e▮▮K▮▮f▮▮▮M▮e▮▮M▮▮fNJP▮▮
   ")
XU 180 B$="▮":B$(128)=B$:B$(C2)=B$:BADR=A
   DR(B$)
GI 190 G$="▮":G$(616)=G$:G$(C2)=G$
KR 200 IOCB=832+16:ADRHI=INT(ADR(G2$)/256
   ):ADRLO=ADR(G2$)-ADRHI*256
GH 260 GOSUB 31000:TRAP OFF:GOSUB 1630:PO
   SITION C2,C3
MT 270 ? "  ▯▯▯":? "  ▯■▯ CREATE ANIMATIO
   N FILE":? "  ▯▯▯"
DQ 280 ? "  ▯▯▯":? "  ▯ ▯ LOAD ANIMATION
   FILE":? "  ▯▯▯"
ST 290 ? "  ▯▯▯":? "  ▯ ▯ SAVE ANIMATION
   FILE":? "  ▯▯▯"
VL 300 ? "  ▯▯▯":? "  ▯ ▯ SEQUENCER":? "
    ▯▯▯"
KO 310 ? "  ▯▯▯":? "  ▯ ▯ VIEW ANIMATION"
   :? "  ▯▯▯"
HO 320 ? "  ▯▯▯":? "  ▯ ▯ SCRIPTER":? "
   ▯▯▯"
XK 330 GOSUB 1680
WD 340 OLDX=C5:OLDY=C4:CURX=OLDX:CURY=OLD
   Y
XO 350 CLOSE #C1:OPEN #C1,C4,C0,"K:":GET
   #C1,KEY
MJ 360 IF KEY=45 AND CURY>C4 THEN CURY=CU
   RY-C3:GOTO 450
UQ 370 IF KEY=61 AND CURY<19 THEN CURY=CU
   RY+C3:GOTO 450
XS 380 IF KEY=155 AND CURY=C4 THEN 470
PP 390 IF KEY=155 AND CURY=C7 THEN 1460
RI 400 IF SEQ$="" THEN 440
LJ 410 IF KEY=155 AND CURY=10 THEN 1330
RF 420 IF KEY=155 AND CURY=13 THEN 1170
NC 430 IF KEY=155 AND CURY=16 THEN 960
FI 440 IF KEY=155 AND CURY=19 THEN TRAP 2
   60:RUN "D:SCRIPTER.BAS"
QO 450 POSITION OLDX,OLDY:? " ";:POSITION
    CURX,CURY:? "■";:OLDX=CURX:OLDY=CURY
US 460 GOTO 350
DO 470 GOSUB 1630:GOSUB 1680:POKE 752,C0
SD 480 ? :? "  How many pictures do you w
   ish to":? "  load (maximum=";MAX;")";:
   INPUT KEY$:IF KEY$="" THEN 260
EH 490 FRAMES=VAL(KEY$):SEQ$="":FOR I=C1
   TO FRAMES:SEQ$(I,I)=CHR$(I):NEXT I:IF
   FRAMES>MAX THEN FRAMES=MAX
VI 500 POKE 752,C1:POSITION C4,22:? "INSE
   RT GRAPHICS DISK, PRESS RETURN":CLOSE
   #C1:OPEN #C1,C4,C0,"K:":GET #C1,KEY
ZW 510 X=USR(SRADR,BADR,361,C1,C3):IF B$(
   C1,15)="PRINT SHOP:CLK!" THEN 530
FN 520 POSITION C7,C8:? "* NOT■PRINT■SHOP
   ■FORMAT *▮";:FOR I=C1 TO 300:NEXT I:GO
   TO 470
XZ 530 GOSUB 1690
ZJ 540 L=ADR(G$):SCRMEM=PEEK(88)+PEEK(89)
   *256+C4+C4*20
DU 560 REM LOAD ICONS
CS 570 ICON=C0
KU 580 FOR SECTR1=362 TO 393:X=USR(SRADR,
   BADR,SECTR1,C1,C3)
TN 590 FOR I=C0 TO 96 STEP 32:REM If B$(X
   +19,X+19)<>"X" And B$(X+20,X+20)<>"x"
   Then Pop :Pop :Goto 570
XZ 600 ICON=ICON+C1
NI 610 SECTR=ASC(B$(I+17,I+17))+256*ASC(B
   $(I+18,I+18))
DX 620 FOR I2=C0 TO 378 STEP 126:X=USR(SR
   ADR,BADR,SECTR,C1,C3)
XR 630 G$(I2+C1,I2+126)=B$(C1,126):SECTR=
   ASC(B$(127,127))+256*ASC(B$(128,128)):
   NEXT I2
WK 640 X=USR(SRADR,BADR,SECTR,C1,C3):G$(5
   05,572)=B$(C1,68)
BL 650 X=USR(SRADR,BADR,SECTR1,C1,C3)
PF 660 X=USR(PICT,SCRMEM,L,11,52,C20)
SU 670 G2$(C1+616*(ICON-C1))=G$
MF 680 IF ICON=FRAMES THEN 970
UL 690 NEXT I:NEXT SECTR1
ZV 950 REM DISPLAY IMAGES
YJ 960 GOSUB 1690
ZT 970 L=ADR(G$):SCRMEM=PEEK(88)+PEEK(89)
   *256+C4+C4*20
HT 980 FOR I=C1 TO LEN(SEQ$)
KJ 990 IF ASC(SEQ$(I,I))=C0 THEN GOSUB 15
   90:GOTO 1020
DO 1000 L=ADR(G2$)+616*(ASC(SEQ$(I,I))-C1
   )
SK 1010 X=USR(PICT,SCRMEM,L,11,52,C20)
XD 1020 FOR DELAY=C1 TO WAIT:IF PEEK(764)
   <>255 THEN GOSUB 1060
II 1025 NEXT DELAY:POKE 77,C0
PQ 1030 IF PEEK(764)<>255 THEN GOSUB 1060
FB 1040 NEXT I
IN 1050 POKE 764,255:GOTO 980
VF 1060 CLOSE #C1:OPEN #C1,C4,C0,"K:":GET
   #C1,KEY
EW 1070 IF KEY=155 AND CURX=C4 THEN WAIT=
   WAIT+C1+C4*(WAIT>29)+C5*(WAIT>45)+10*(
   WAIT>90)
JJ 1080 IF KEY=155 AND CURX=11 THEN WAIT=
   WAIT-C1-C4*(WAIT>30)-C5*(WAIT>50)-10*(
   WAIT>100):IF WAIT<C0 THEN WAIT=C0
WZ 1090 IF KEY=155 AND CURX=18 THEN PAUSE
   = NOT PAUSE
KN 1100 IF KEY=155 AND CURX=25 AND PAUSE
   THEN 1270
DH 1110 IF KEY=155 AND CURX=32 THEN POP :
   POP :GOTO 260
VL 1120 IF KEY=42 AND CURX<32 THEN CURX=C
   URX+C7:POKE 657,OLDX:? " ";:POKE 657,C
   URX:? "■";:OLDX=CURX
SO 1130 IF KEY=43 AND CURX>C4 THEN CURX=C
   URX-C7:POKE 657,OLDX:? " ";:POKE 657,C
   URX:? "■";:OLDX=CURX
FI 1140 IF PAUSE THEN 1060
AQ 1150 RETURN
AJ 1160 GOSUB 1060
WC 1170 REM SEQUENCER
```

continued on next page

```
UG 1180 GOSUB 1630:GOSUB 1680
ER 1190 ? :? "ENTER '0' TO PAUSE SEQUENCE
   ."
VY 1200 FRAME=C1:SEQ$="":?
NH 1210 ? "FRAME #";FRAME;":KKKKK";
CO 1220 INPUT KEY$:IF KEY$="" AND SEQ$=""
    THEN SEQ$=CHR$(C1):GOTO 960
ZM 1230 IF KEY$="" THEN FRAME=FRAME-C1:GO
   TO 960
ZJ 1240 KEY=VAL(KEY$):IF KEY<C0 OR KEY>FR
   AMES THEN ? "ᴿ";:GOTO 1210
RG 1250 SEQ$(FRAME,FRAME)=CHR$(KEY)
PA 1260 FRAME=FRAME+C1:IF FRAME<101 THEN
   1210
TX 1265 FRAME=FRAME-C1:GOTO 960
IZ 1270 REM ADVANCE
LF 1280 I=I+C1:IF I>LEN(SEQ$) THEN I=C1
AL 1290 IF ASC(SEQ$(I,I))=C0 THEN 1280
DU 1300 L=ADR(G2$)+616*(ASC(SEQ$(I,I))-C1
   )
SQ 1310 X=USR(PICT,SCRMEM,L,11,52,C20)
QA 1320 GOTO 1060
WW 1325 REM SAVE SEQUENCE FILE
CI 1330 GOSUB 1630:GOSUB 1680:TRAP 3000
XO 1340 POSITION C2,C3:? "USE FORMAT: 'DE
   V:FILENAME.EXT'"
WM 1350 POKE 752,C0:POSITION C2,C5:? "FIL
   ENAME TO SAVE?";:POKE 764,255
FI 1360 ON PEEK(764)=255 GOTO 1360:IF PEE
   K(764)=28 THEN CLOSE #C1:OPEN #C1,C4,C
   0,"K:":GET #C1,KEY:GOTO 260
QH 1370 ? "";:INPUT FILENAME$:IF FILENAM
   E$="" THEN GOSUB 2320:GOTO 1350
FP 1380 CLOSE #C1:OPEN #C1,C8,C0,FILENAME
   $
HM 1390 SEQ$(LEN(SEQ$)+C1)=CHR$(FRAMES):?
    #C1;SEQ$
VB 1400 POKE IOCB+C2,11
IG 1410 POKE IOCB+C4,ADRLO:POKE IOCB+C5,A
   DRHI
WZ 1420 NM=616*FRAMES:NMHI=INT(NM/256):NM
   LO=NM-NMHI*256
XX 1430 POKE IOCB+C8,NMLO:POKE IOCB+C9,NM
   HI
VW 1440 X=USR(ADR("hhhLVd"),16)
LO 1450 CLOSE #C1:SEQ$=SEQ$(C1,LEN(SEQ$)-
   C1):GOTO 260
QI 1455 REM LOAD SEQUENCE FILE
CT 1460 GOSUB 1630:GOSUB 1680:TRAP 3000
AS 1470 POSITION C2,C3:? "USE FORMAT: 'DE
   V:FILENAME.EXT'":SEQ$=""
UA 1480 POKE 752,C0:POSITION C2,C5:? "FIL
   ENAME TO LOAD?";:POKE 764,255
KI 1490 ON PEEK(764)=255 GOTO 1490:IF PEE
   K(764)=28 THEN CLOSE #C1:OPEN #C1,C4,C
   0,"K:":GET #C1,KEY:GOTO 260
AB 1500 ? "";:INPUT FILENAME$:IF FILENAM
   E$="" THEN GOSUB 2320:GOTO 1480
AY 1510 CLOSE #C1:OPEN #C1,C4,C0,FILENAME
   $
QG 1520 INPUT #C1;SEQ$:FRAMES=ASC(SEQ$(LE
   N(SEQ$),LEN(SEQ$))):SEQ$=SEQ$(C1,LEN(S
   EQ$)-C1)
NG 1530 POKE IOCB+C2,C7
IR 1540 POKE IOCB+C4,ADRLO:POKE IOCB+C5,A
   DRHI
XK 1550 NM=616*FRAMES:NMHI=INT(NM/256):NM
   LO=NM-NMHI*256
YI 1560 POKE IOCB+C8,NMLO:POKE IOCB+C9,NM
   HI
WH 1570 X=USR(ADR("hhhLVd"),16)
KO 1580 CLOSE #C1:GOTO 960
ON 1590 REM PAUSE
AO 1600 FOR DELAY=C1 TO 100
PW 1610 IF PEEK(764)<>255 THEN GOSUB 1060
JL 1620 NEXT DELAY:RETURN
NM 1630 REM TITLE SCREEN
EM 1640 GRAPHICS C0:POKE 709,C0:POKE 710,
   C0:GOSUB 31000:POKE 752,C1
LN 1650 POSITION C0,C0:? "Icon Animation
   System    * SEQUENCER *  "
LP 1660 POSITION C0,C1:? "===============
   ========================"
BG 1670 RETURN
ZM 1680 FOR I=C1 TO 10:POKE 710,I:NEXT I:
   RETURN
EF 1690 GRAPHICS MODE:GOSUB 31000:POKE 75
   2,C1:PAUSE=C0
YI 1700 POKE 708,C0:POKE 709,C0:POKE 710,
   C0:POKE 712,C0
YX 1710 COLOR C1:PLOT 30,C2:DRAWTO 121,C2
   :DRAWTO 121,57:DRAWTO 30,57:DRAWTO 30,
   C2
AB 1720 ? "                              
   "
GG 1730 ? "  SLOW  FAST  PAUS  ADV. 
   EXIT"
FV 1740 ? "                              
   "
WS 1750 OLDX=C4:OLDY=C1:CURX=OLDX:CURY=OL
   DY
BF 1760 POKE 656,CURY:POKE 657,CURX:? ""
   ;
PV 1770 FOR I=C1 TO 10:POKE 710,I:POKE 71
   2,I:NEXT I:RETURN
BL 1780 RETURN
HF 2320 REM DIRECTORY
NX 2340 G$(C1)=" ":G$(616)=" ":G$(C2)=G$:
   K=C0:L=C0:POKE 752,C1
NV 2350 CLOSE #C1:OPEN #C1,C6,C0,"D1:*.*"
CO 2360 INPUT #C1,FILENAME$
NQ 2370 IF FILENAME$(C4,11)="SHOP:CLK" TH
   EN GOSUB 2440:RETURN
NK 2380 IF FILENAME$(C5,C8)="FREE" THEN C
   LOSE #C1:GOTO 2410
WI 2390 G$(17*K+C1,17*K+LEN(FILENAME$))=F
   ILENAME$:K=K+C1
RS 2400 GOTO 2360
ZO 2410 FOR I=C0 TO 13:POSITION C1,C7+I:?
    G$(17*(L+I)+C1,17*(L+I)+17);"  ";G$(1
   7*(L+I+14)+C1,17*(L+I+14)+17)
NC 2420 NEXT I:? :? FILENAME$;"
   ";
AR 2430 RETURN
OQ 2440 POSITION 2,22:? "        * PRINT S
   HOP FORMAT *";:RETURN
HA 3000 REM ERROR MSG
HY 3010 CLOSE #C1:POKE 703,C4:POKE 752,C1
   :? "":? "      * FILE ACCESS ERROR
   *    ";
ZA 3020 FOR I=C1 TO 100:NEXT I:POKE 703,2
   4:GOTO 260
JS 31000 BRK=USR(ADR("h MhhPpR
   "),C1):RETURN :REM DISABLES BREAK WHEN
    ARG=1
```

LISTING 2

```
ZR 100 REM ICON ANIMATION SYSTEM,
       LISTING 2, THE SCRIPTER
BO 103 REM BY JIM JOHNSON
HM 106 REM (c)1989 ANTIC PUBLISHING INC.
VQ 110 C0=0:C1=1:C2=2:C3=3:C4=4:C5=5:C6=6
   :C7=7:C8=8:C9=9:C20=20:MAX=C20+C5
KD 120 MODE=C6:OFF=40000:WAIT=10:GOSUB 31
   000:GOSUB 30000
EG 130 DIM FILENAME$(20),TEMP$(20),DEVICE
   $(3),DPATH$(6)
NT 140 DIM G$(616),G2$(616*MAX)
OC 150 DIM SEQ$(101),FILE$(20*10),SCRIPT(
   101),CYCLE(10),FNL(10)
YX 170 PICT=ADR("hhLhKhNhhhhhhP
   S0ZMKePWKePKFLMeMFNJPC
   ")
HJ 190 G$="":G$(616)=G$:G$(2)=G$
XZ 260 TRAP OFF:GOSUB 1500:POSITION C2,C3

OC 270 ? "    ":? "    CREATE SCRIPT":
   ? "    "
TF 280 ? "    ":? "    LOAD SCRIPT":?
   "    "
II 290 ? "    ":? "    SAVE SCRIPT":?
   "    "
QZ 300 ? "    ":? "    RUN SCRIPT":? "
   "
VN 310 ? "    ":? "    SEQUENCER":? "
   "
WE 320 ? "    ":? "    LOAD SEQUENCES
   TO RAMDISK":? "    "
CE 330 IF RAMDISK=1 THEN POSITION 5,19:?
   "X":?
VN 340 GOSUB 1550
WF 350 OLDX=C5:OLDY=C4:CURX=OLDX:CURY=OLD
   Y
XQ 360 CLOSE #C1:OPEN #C1,C4,C0,"K:":GET
   #C1,KEY
OK 370 IF KEY=45 AND CURY>C4 THEN CURY=CU
   RY-C3:GOTO 460
UQ 380 IF KEY=61 AND CURY<19-(3*(RAMDISK=
   1)) THEN CURY=CURY+C3:GOTO 460
ZC 390 IF KEY=155 AND CURY=C4 THEN 480
CR 400 IF KEY=155 AND CURY=C7 THEN 670
PY 410 IF KEY=155 AND CURY=10 AND FILE$<>
   "" THEN 740
```

```
NL 420 IF KEY=155 AND CURY=13 AND FILE$<>"" THEN 800
WK 430 IF KEY=155 AND CURY=16 THEN TRAP 260:RUN "D:ANIMATOR.BAS"
NP 440 IF KEY=155 AND CURY=19 THEN 850
QQ 460 POSITION OLDX,OLDY:? " ";:POSITION CURX,CURY:? "■";:OLDX=CURX:OLDY=CURY
PF 470 GOTO 360
AZ 480 REM CREATE SCRIPT
HW 490 GOSUB 1500:GOSUB 1550
GC 500 DEVICE$="D?:":DEVICE$(2,2)=STR$(RAMDISK):GOSUB 1700:GOSUB 1820
LM 510 IF  NOT FILE THEN CLOSE #C1:OPEN #C1,C4,C0,"K:":GET #C1,KEY:GOTO 260
UR 520 POSITION 17,C3:? "SEQUENCE  CYCLES";:POSITION 17,C4:? "□□□□□□□□  □□□□□□";
JR 530 OLDX=C3:OLDY=C4+FILE*C2:CURX=OLDX:CURY=OLDY:I=C1
XO 540 CLOSE #C1:OPEN #C1,C4,C0,"K:":GET #C1,KEY
UM 550 IF KEY=45 AND CURY>C4 THEN CURY=CURY-C2:GOTO 590
DO 560 IF KEY=61 AND CURY<C4+FILE*C2 THEN CURY=CURY+C2:GOTO 590
ZC 570 IF KEY=155 AND CURY=C4+FILE*C2 THEN 260
MR 580 IF KEY=155 AND CURY>C3 AND CURY<C5+FILE*C2 THEN 610
QX 590 POSITION OLDX,OLDY:? " ";:POSITION CURX,CURY:? "■";:OLDX=CURX:OLDY=CURY
OR 600 GOTO 540
TO 610 FILE$=G$:FILENAME$=FILE$(C8*((CURY/C2)-C2)+C1,C8*((CURY/C2)-C2)+FNL(CURY/C2-C1))
LH 620 SCRIPT(I)=CURY/C2-C1:POSITION OLDX,OLDY:? " ";
ID 630 POSITION 17,C4+I:? FILENAME$:POSITION 28,C4+I:? "    ⊞⊞⊞⊞";:POKE 752,C0
OK 640 TRAP 630:INPUT KEY:POKE 752,C1:? :IF KEY<C0 OR KEY>100 THEN 630
VD 650 CYCLE(I)=KEY:SEQ=I:I=I+C1:IF I=11 THEN 260
QN 660 POSITION OLDX,OLDY:? "■";:GOTO 540
OY 670 REM LOAD SCRIPT
AZ 680 TRAP 3000:DEVICE$="D?:":DEVICE$(2,2)=STR$(RAMDISK):GOSUB 1700:GOSUB 1500:GOSUB 1550
WB 690 POSITION C2,C3:? "USE FORMAT: 'DEV:FILENAME.EXT'"
TX 695 POKE 752,C0:POSITION C2,C5:? "FILENAME TO LOAD?";:POKE 764,255
IS 696 ON PEEK(764)=255 GOTO 696:IF PEEK(764)=28 THEN CLOSE #C1:OPEN #C1,C4,C0,"K:":GET #C1,KEY:GOTO 260
SL 700 ? "←";:INPUT FILENAME$:IF FILENAME$="" THEN GOSUB 2320:GOTO 695
QA 710 CLOSE #C1:OPEN #C1,C4,C0,FILENAME$
XJ 720 GET #1,SEQ:FOR I=C1 TO SEQ:GET #C1,J:SCRIPT(I)=J:GET #C1,J:CYCLE(I)=J:NEXT I:INPUT #C1;FILE$:CLOSE #C1
EB 725 FOR J=C1 TO SEQ:FILENAME$=FILE$(C8*(J-C1)+C1,C8*(J-C1)+C8):I=C8
VR 727 IF FILENAME$(I,I)=" " THEN I=I-C1:GOTO 727
KP 728 FNL(J)=I:NEXT J
OW 730 GOTO 260
VD 740 REM SAVE SCRIPT
RM 750 TRAP 3000:GOSUB 1500:GOSUB 1550:POSITION 2,3:? "USE FORMAT: 'DEV:FILENAME.EXT'"
VY 755 POKE 752,C0:POSITION C2,C5:? "FILENAME TO SAVE?";:POKE 764,255
FE 756 ON PEEK(764)=255 GOTO 756:IF PEEK(764)=28 THEN CLOSE #C1:OPEN #C1,C4,C0,"K:":GET #C1,KEY:GOTO 260
LG 760 ? "←";:INPUT FILENAME$:IF FILENAME$="" THEN GOSUB 2320:GOTO 755
UI 770 CLOSE #C1:OPEN #C1,C8,C0,FILENAME$
YU 780 PUT #C1,SEQ:FOR I=C1 TO SEQ:J=SCRIPT(I):PUT #C1,J:J=CYCLE(I):PUT #C1,J:NEXT I:? #C1;FILE$:CLOSE #C1
PC 790 GOTO 260
ZS 800 REM RUN SCRIPT
BM 810 FILENAME$="D?:":FILENAME$(2,2)=STR$(RAMDISK):GOSUB 1560:G$=FILE$
CE 820 FOR J=C1 TO SEQ:I=SCRIPT(J):GOSUB 1360
XI 830 FOR K=C1 TO CYCLE(J):GOSUB 970:NEXT K
RG 840 NEXT J:POKE 77,C0:GOTO 820
GY 850 REM LOAD RAMDISK
HU 860 GOSUB 1500:GOSUB 1550
MP 870 DEVICE$="D1:":GOSUB 1700:GOSUB 1820
MD 880 IF  NOT FILE THEN CLOSE #C1:OPEN #C1,C4,C0,"K:":GET #C1,KEY:GOTO 260
HZ 890 IF PEEK(53279)=3 THEN 910
XY 892 POKE 5439,ASC("1"):REM LOAD DUP.SYS FROM DRIVE #1
SB 895 DPATH$="D?:*.*":DPATH$(2,2)=STR$(RAMDISK):TRAP 910:CLOSE #C1:OPEN #C1,C4,C0,DPATH$
SU 900 XIO 33,#C1,C0,C0,DPATH$:CLOSE #C1:GOTO 895
OJ 910 POSITION 21,11:? "*LOADING*":POSITION C3,C4+FILE*C2:? " ";
TX 920 FILENAME$=DEVICE$:FOR I=C1 TO FILE:POSITION C3,C2+I*C2:? "■";
UY 930 FILENAME$="D1:":GOSUB 1360:GOSUB 1240
AW 940 POSITION C3,C2+I*C2:? " ";:NEXT I:GOTO 260
ZV 950 REM DISPLAY IMAGES
WK 960 GOSUB 1560
ZT 970 L=ADR(G$):SCRMEM=PEEK(88)+PEEK(89)*256+C4+C4*20
HT 980 FOR I=C1 TO LEN(SEQ$)
JR 990 IF ASC(SEQ$(I,I))=C0 THEN GOSUB 1470:GOTO 1040
DO 1000 L=ADR(G2$)+616*(ASC(SEQ$(I,I))-C1)
SK 1010 X=USR(PICT,SCRMEM,L,11,52,C20)
XD 1020 FOR DELAY=C1 TO WAIT:IF PEEK(764)<>255 THEN GOSUB 1060
II 1025 NEXT DELAY:POKE 77,C0
PQ 1030 IF PEEK(764)<>255 THEN GOSUB 1060
FB 1040 NEXT I
AO 1050 RETURN
JJ 1060 CLOSE #C1:OPEN #C1,C4,C0,"K:":GET #C1,KEY:POKE 764,255
EW 1070 IF KEY=155 AND CURX=C4 THEN WAIT=WAIT+C1+C4*(WAIT>29)+C5*(WAIT>45)+10*(WAIT>90)
JJ 1080 IF KEY=155 AND CURX=11 THEN WAIT=WAIT-C1-C4*(WAIT>30)-C5*(WAIT>50)-10*(WAIT>100):IF WAIT<C0 THEN WAIT=C0
WZ 1090 IF KEY=155 AND CURX=18 THEN PAUSE= NOT PAUSE
IU 1100 IF KEY=155 AND CURX=25 AND PAUSE THEN 1170
DH 1110 IF KEY=155 AND CURX=32 THEN POP :POP :GOTO 260
VL 1120 IF KEY=42 AND CURX<32 THEN CURX=CURX+C7:POKE 657,OLDX:? " ";:POKE 657,CURX:? "■";:OLDX=CURX
KF 1130 IF KEY=43 AND CURX>4 THEN CURX=CURX-C7:POKE 657,OLDX:? " ";:POKE 657,CURX:? "■";:OLDX=CURX
FI 1140 IF PAUSE THEN 1060
AQ 1150 RETURN
AJ 1160 GOSUB 1060
IX 1170 REM ADVANCE
LC 1175 IF SEQ$="" THEN 1060
LD 1180 I=I+C1:IF I>LEN(SEQ$) THEN I=C1
ZC 1190 IF ASC(SEQ$(I,I))=C0 THEN 1180
DS 1200 L=ADR(G2$)+616*(ASC(SEQ$(I,I))-C1)
SO 1210 X=USR(PICT,SCRMEM,L,11,52,C20)
PY 1220 GOTO 1060
ZJ 1230 GOSUB 1500
WG 1240 REM SAVE SEQUENCE FILE
TM 1250 FILENAME$(C2,C2)="8"
YG 1260 CLOSE #C1:TRAP 1345:OPEN #C1,C8,C0,FILENAME$
HE 1270 SEQ$(LEN(SEQ$)+C1)=CHR$(FRAMES):? #C1;SEQ$
RR 1280 IOCB=832+16:POKE IOCB+2,11
LY 1290 ADRHI=INT(ADR(G2$)/256):ADRLO=ADR(G2$)-ADRHI*256
IB 1300 POKE IOCB+C4,ADRLO:POKE IOCB+C5,ADRHI
WU 1310 NM=616*FRAMES:NMHI=INT(NM/256):NMLO=NM-NMHI*256
XS 1320 POKE IOCB+C8,NMLO:POKE IOCB+C9,NMHI
DI 1330 X=USR(ADR("hhh⊠LV⊡"),16):IF (PEEK(IOCB+C9)*256+PEEK(IOCB+C8))<>NRED THEN 1345
XG 1340 CLOSE #C1:SEQ$=SEQ$(C1,LEN(SEQ$)-C1):RETURN
QI 1345 TRAP 260:CLOSE #C1:OPEN #C1,C4,C0,FILENAME$:XIO 33,#C1,C0,C0,FILENAME$:CLOSE #C1:GOTO 260
```

continued on next page

```
ZR 1350 GOSUB 1500
PP 1360 REM LOAD SEQUENCE FILE
FM 1370 FILENAME$(C4)=G$(C8*(I-C1)+C1,C8*
   (I-C1)+FNL(I)):FILENAME$(C4+FNL(I))=".
   SEQ"
XY 1380 TRAP 1465:CLOSE #C1:OPEN #C1,C4,C
   0,FILENAME$
QX 1390 INPUT #C1;SEQ$:FRAMES=ASC(SEQ$(LE
   N(SEQ$),LEN(SEQ$))):SEQ$=SEQ$(C1,LEN(S
   EQ$)-C1)
VH 1400 IOCB=832+16:POKE IOCB+C2,C7
LE 1410 ADRHI=INT(ADR(G2$)/256):ADRLO=ADR
   (G2$)-ADRHI*256
IJ 1420 POKE IOCB+C4,ADRLO:POKE IOCB+C5,A
   DRHI
XC 1430 NM=616*FRAMES:NMHI=INT(NM/256):NM
   LO=NM-NMHI*256
YA 1440 POKE IOCB+C8,NMLO:POKE IOCB+C9,NM
   HI
PG 1450 X=USR(ADR("hhh█LV█"),16):NRED=PEE
   K(IOCB+C9)*256+PEEK(IOCB+C8)
ID 1460 CLOSE #C1:RETURN
SW 1465 POKE 656,C3:POKE 657,C2:? "* MISS
   ING█FILE * (";FILENAME$;")    ";:POKE 6
   56,CURY
YV 1466 FOR DELAY=C1 TO WAIT:IF PEEK(764)
   <>255 THEN GOSUB 1060
JK 1467 NEXT DELAY:POKE 77,C0
FV 1468 POP :GOTO 840
OF 1470 REM PAUSE
DC 1480 FOR DELAY=C1 TO 100:IF PEEK(764)<
   >255 THEN GOSUB 1060
KC 1490 NEXT DELAY:RETURN
NB 1500 REM TITLE SCREEN
EB 1510 GRAPHICS C0:POKE 709,C0:POKE 710,
   C0:GOSUB 31000:POKE 752,C1
XU 1520 POSITION C0,C0:? "█Icon█Animation
   █System███* Scripter *███"
LE 1530 POSITION C0,C1:? "███████████████
   ██████████████████████████"
AV 1540 RETURN
ZB 1550 FOR I=C1 TO 10:POKE 710,I:NEXT I:
   RETURN
DU 1560 GRAPHICS MODE:GOSUB 31000:POKE 75
   2,C1:PAUSE=C0
YZ 1570 POKE 708,C0:POKE 709,C0:POKE 710,
   C0:POKE 712,C0
ZO 1580 COLOR C1:PLOT 30,C2:DRAWTO 121,C2
   :DRAWTO 121,57:DRAWTO 30,57:DRAWTO 30,
   C2
AS 1590 ? " ┌──┐     ┌──┐     ┌──┐     ┌──┐     ┌
   ──┐"
FV 1600 ? " │ │SLOW│ │FAST│ │PAUS│ │ADV.│
   │EXIT"
FK 1610 ? " └──┘     └──┘     └──┘     └──┘     └
   ──┘"
WH 1620 OLDX=C4:OLDY=C1:CURX=OLDX:CURY=OL
   DY
AU 1630 POKE 656,CURY:POKE 657,CURX:? "█"
   ;
PK 1640 FOR I=C1 TO 10:POKE 710,I:POKE 71
   2,I:NEXT I:RETURN
BA 1650 RETURN
BG 1660 FOR DELAY=C1 TO 100
QO 1670 IF PEEK(764)<>255 THEN GOSUB 1060
KD 1680 NEXT DELAY:RETURN
BM 1690 RETURN
ZZ 1700 REM DIRECTORY OF SEQUENCES
JS 1710 G$(C1)=" ":G$(20*10)=" ":G$(C2)=G
   $:FILE=C0
SP 1720 FILENAME$=DEVICE$:FILENAME$(4)="*
   .SEQ":CLOSE #C1:OPEN #C1,C7,C0,FILENAM
   E$
CM 1730 INPUT #C1,FILENAME$
MC 1740 IF FILENAME$(C5,C8)="FREE" OR FIL
   E>8 THEN CLOSE #C1:GOTO 1800
EW 1750 FILE=FILE+C1
MM 1760 FILENAME$=FILENAME$(C3,LEN(FILENA
   ME$)-C7):G$(C8*(FILE-C1)+C1,C8*(FILE-C
   1)+C8)=FILENAME$
KE 1770 FOR I=C8 TO C1 STEP -C1:IF FILENA
   ME$(I,I)<>" " THEN FNL(FILE)=I:POP :GO
   TO 1790
GB 1780 NEXT I
SW 1790 GOTO 1730
AP 1800 RETURN
KL 1820 ? "┌────────────┐":IF  NOT FILE THE
   N 1840
HD 1830 FOR I=C1 TO FILE:? "│ │";:? G$(C8
   *(I-C1)+C1,C8*(I-C1)+C8);:? "│":? "├──
       ┤":NEXT I
LP 1840 ? "│██EXIT███│";:IF FILE THEN ?
   :GOTO 1860
MC 1850 ? "  * NO SEQUENCE FILES *"
RF 1860 ? "└────────────┘";
BK 1870 RETURN
HF 2320 REM DIRECTORY
NX 2340 G$(C1)=" ":G$(616)=" ":G$(C2)=G$:
   K=C0:L=C0:POKE 752,C1
OV 2350 CLOSE #C1:OPEN #C1,C7,C0,"D1:*.*"
CO 2360 INPUT #C1,FILENAME$
NQ 2370 IF FILENAME$(C4,11)="SHOP:CLK" TH
   EN GOSUB 2440:RETURN
NK 2380 IF FILENAME$(C5,C8)="FREE" THEN C
   LOSE #C1:GOTO 2410
WI 2390 G$(17*K+C1,17*K+LEN(FILENAME$))=F
   ILENAME$:K=K+C1
RS 2400 GOTO 2360
ZO 2410 FOR I=C0 TO 13:POSITION C1,C7+I:?
    G$(17*(L+I)+C1,17*(L+I)+17);"  ";G$(1
   7*(L+I+14)+C1,17*(L+I+14)+17)
NC 2420 NEXT I:? :? FILENAME$;"
     ";
AR 2430 RETURN
OU 2440 POSITION 2,22:? "        * PRINT█S
   HOP█FORMAT *█";:RETURN
HA 3000 REM ERROR MSG
SQ 3010 POKE 703,C4:POKE 752,C1:? "█":? "
        * █FILE█ACCESS█ERROR█ *      █";
ZA 3020 FOR I=C1 TO 100:NEXT I:POKE 703,2
   4:GOTO 260
OX 30000 RAMDISK=1:TRAP 30010:CLOSE #4:OP
   EN #4,6,0,"D8:*.*":RAMDISK=8
QL 30010 TRAP 30020:CLOSE #4
KS 30020 TRAP OFF:RETURN
ZL 31000 BRK=USR(ADR("h█♥█Mhh█♥█N█p█♣█_R♦
   "),C1):RETURN
```

LISTING 3

```
IN 10 REM ICON ANIMATION SYSTEM,
      LISTING 3
YO 20 REM BY JIM JOHNSON
GX 30 REM (c) 1985,1989 ANTIC PUBLISHING
ZD 35 REM (CREATES LINES 170 & 31000 FOR
   LISTINGS 1 AND 2. CREATES LINE 160 FOR
    LISTING 1)
EV 40 REM (LINES 10-250 MAY BE USED WITH
   OTHER BASIC LOADERS IN THIS ISSUE.
IJ 50 REM CHANGE LINE 70 AS NECESSARY.)
PR 60 DIM FN$(20),TEMP$(20),AR$(93):DPL=P
   EEK(10592):POKE 10592,255
WO 70 FN$="D:LINES.LST":REM THIS IS THE N
   AME OF THE DISK FILE TO BE CREATED
RD 80 ? "█Disk or █assette?";:POKE 764,25
   5
PY 90 IF  NOT (PEEK(764)=18 OR PEEK(764)=
   58) THEN 90
TH 100 IF PEEK(764)=18 THEN FN$="C:"
VB 110 POKE 764,255:GRAPHICS 0:? "      AN
   TIC'S GENERIC BASIC LOADER"
MY 120 ? ,"BY CHARLES JACKSON"
KB 130 POKE 10592,DPL:TRAP 200
PU 140 ? :? :? "Creating ";FN$:? "...plea
   se stand by."
LW 150 RESTORE :READ LN:LM=LN:DIM A$(LN):
   C=1
BQ 160 AR$="":READ AR$
YC 170 FOR X=1 TO LEN(AR$) STEP 3:POKE 75
   2,255
DM 180 LM=LM-1:POSITION 10,10:? "(Countdo
   wn...T-";INT(LM/10);")      "
BK 190 A$(C,C)=CHR$(VAL(AR$(X,X+2))):C=C+
   1:NEXT X:GOTO 160
MM 200 IF PEEK(195)=5 THEN ? :? :? "█TOO
   MANY DATA LINES!":? "CANNOT CREATE FIL
   E!":END
CM 210 IF C<LN+1 THEN ? :? "█TOO FEW DATA
    LINES!":? "CANNOT CREATE FILE!":END
UQ 220 IF FN$="C:" THEN ? :? " Prepare ca
   ssette, press [RETURN]"
AR 230 OPEN #1,8,0,FN$
PV 240 POKE 766,1:? #1;A$;:POKE 766,0
AL 250 CLOSE #1:GRAPHICS 0:? "█COMPLETED█
   "
FV 1000 DATA 204
DW 1010 DATA 0490540480320830820650680820
   61065068082040034104104141005003104141
   004003104141011003104141010
UL 1020 DATA 0031041041410010031041041701
   69082141002003032083228202173003003201
   001240004224000208241169000
DL 1030 DATA 133213173003003133212096034
   0411550490550480320800730670840610650
   68082040034104104133204104133
UJ 1040 DATA 2031041332061041332051041041
```

```
   3320710410417010410413320816420713617
   7205145203136192255208247024
BN 1050 DATA 165203101208133203144002230
   2040241652051012071332051440022302062
   02208219096034041155051049048
XK 1060 DATA 048048032066082075061085083
   0820400650680820400341041690001330771
   04104201000240007169112133016
RU 1070 DATA 141014210096034041044049041
   05808206908408508207803215 5
```

HELP files printed on half the pages

Two-Column Print Pro

Article on page 13

LISTING 1

Don't type the TYPO II Codes!

```
DA 2 REM TWO-COLUMN PRINTING
AD 4 REM BY EARL HALEY
PF 6 REM (c)1989, ANTIC PUBLISHING INC.
LF 10 GRAPHICS 0:POKE 710,14:POKE 712,14:
   POKE 709,2:DIM BUZ$(3):FOR N=1 TO 3:BU
   Z$(N)=CHR$(253):NEXT N
GJ 20 ? :? :? " TWO-COLUMN PRINTER PROGRA
   M":? :? "  By Earl Haley"
AE 25 KSIZE=40*80
FH 30 DIM BLANK$(KSIZE),COL1$(KSIZE),COL2
   $(KSIZE),R$(41),TEST$(2),FILENAME$(15)
   :PAGE=0:FILE=1:FINISHED=0
KS 40 BLANK$(1)=" ":BLANK$(KSIZE)=" ":BLA
   NK$(2)=BLANK$:COL1$=BLANK$:COL2$=BLANK
   $
KM 50 GOSUB 700:GOSUB 800:GOTO 370
DF 100 TRAP 230:CLOSE #1:OPEN #1,4,0,FILE
   NAME$:PAGE=0
LJ 110 TRAP 230
GL 120 FOR COLUMN=1 TO 2
NF 130 S=1:F=40
BF 140 FOR LINE=1 TO NL:REM (NL DEFINED I
   N LINES 740 AND 750)
FW 150 INPUT #1,R$:IF LEN(R$)=3 THEN IF R
   $(1,3)="XXX" OR R$(1,3)=BUZ$ THEN 150
OO 160 ON COLUMN GOTO 170,180
QQ 170 COL1$(S,F)=R$:GOTO 190
VE 180 COL2$(S,F)=R$
TB 190 S=F+1:F=S+39
EG 200 NEXT LINE
SY 210 NEXT COLUMN
NY 220 GOTO 250
LU 230 IF PEEK(195)=170 THEN ? CHR$(125):
   ? :? "FILE NOT FOUND":CLOSE #1:GOTO 38
   0
WG 235 IF PEEK(195)=160 THEN ? "DRIVE# ER
   ROR":CLOSE #2:GOTO 440
ZZ 240 IF PEEK(195)=144 THEN ? CHR$(125):
   ? :? "DISK DRIVE ERROR":CLOSE #1:GOTO
   380
IH 245 IF PEEK(195)=136 THEN FINISHED=1:G
   OTO 250
RJ 246 ? "DISK ERROR":CLOSE #1:GOTO 380
ZH 250 PAGE=PAGE+1
HD 260 CLOSE #2:OPEN #2,8,0,"P:"
NO 270 S=1:F=40
TI 280 FOR PRYNTLYNE=1 TO NL
AI 290 ? #2;COL1$(S,F);COL2$(S,F)
SL 300 S=F+1:F=S+39
BX 310 NEXT PRYNTLYNE
IA 320 COL1$=BLANK$:COL2$=BLANK$
YH 330 ? #2:? #2;BLANK$(1,36);"- ";PAGE;"
    -"
OH 340 IF FILE<>FINISHED THEN 345
PJ 341 ? #2;BLANK$(1,35);"- END -":FINISH
   ED=0:? "■DONE !":? "Press a Key":GOSUB
    660:RUN
RG 345 IF CP<>1 THEN 350
KT 346 ? "Pause for Paper Change - Check
   Your  Pprinter- Press Any  Key When R
   eady":GOSUB 660:GOTO 110
HW 350 IF CP=2 THEN FOR N=1 TO FS-NL-2:?
   #2:NEXT N
MF 360 GOTO 110
JE 370 CLOSE #2:GRAPHICS 0:POKE 710,50:?
   CHR$(125)
NP 375 IF FS=84 THEN ? :? "< LEGAL SIZE 8
    1/2 x 14 INCHES >"
IH 376 IF FS=66 THEN ? :? "< STANDARD SIZ
   E 8 1/2 x 11 INCHES >"
PP 380 ? :? "Be Sure ANTIC DISK Is In Dri
   ve"
XK 382 ? :? "Be sure Printer and Paper ar
   e ready"
AH 385 ? :? :? "What File do you wish to
   print"
NO 390 ? :? :? "(1)  ANTIC HELP FILE"
QQ 400 ? :? "(2)  OTHER":POKE 752,1:? :?
   "?"
VO 410 GOSUB 660
UG 420 ON BYTE-48 GOTO 430,435
AH 430 FILENAME$="D:DOC.TXT":FLAG=2:GOTO
   100
BQ 435 ? CHR$(125)
EW 440 ? :? :? :? "(Press  RETURN  for di
   sk directory)":? "(Lists only files wi
   th .TXT extender)"
DO 450 ? :? "ENTER NAME OF 40-COLUMN TEXT
    FILE"
EO 460 ? "(D:XXXXXXXX.XXX)"
UO 470 TRAP 230:INPUT FILENAME$:IF FILENA
   ME$="" THEN GOSUB 590:GOTO 440
EW 472 IF LEN(FILENAME$)<3 THEN ? "FILENA
VA 475 TEST$=FILENAME$(1,2)
EG 476 IF TEST$<>"D:" AND TEST$<>"D1" AND
    TEST$<>"D2" AND TEST$<>"D8" THEN ? "D
   RIVE #?":GOTO 450
ML 480 IF LEN(FILENAME$)>15 THEN ? "FILEN
   AME TOO LONG !":? :GOTO 450
SX 490 CLOSE #1:OPEN #1,4,0,FILENAME$
AV 500 TRAP 550:FOR N=1 TO 10
TR 510 INPUT #1,R$:IF LEN(R$)>40 THEN ? "
   NOT A 40-COLUMN FILE":POP :CLOSE #1:GO
   TO 560
VN 520 IF LEN(R$)>1 THEN CLOSE #1:FLAG=2:
   POP :? :GOTO 100
HZ 530 NEXT N
QD 550 CLOSE #1:? :? :? "NOT A 40-COLUMN
   TEXT FILE !"
UO 560 ? :? "PRESS ANY KEY TO CONTINUE"
WB 570 GOSUB 660
QE 580 GOTO 380
TZ 590 TRAP 630:CLOSE #1:OPEN #1,7,0,"D:*
   .*":TEST=0:FOR N=1 TO 99:INPUT #1,R$
PI 600 IF R$(11,13)="TXT" THEN TEST=TEST+
   1:? R$;"  ";
HW 610 NEXT N
GD 630 IF PEEK(195)=136 THEN 650
FO 640 ? "DISK ERROR"
VM 650 IF TEST<1 THEN ? :? "NO 40 COLUMN
   TEXT FILES FOUND":?
IB 655 CLOSE #1:RETURN
CQ 660 CLOSE #4:OPEN #4,4,0,"K:":GET #4,B
   YTE:CLOSE #4:RETURN
GA 700 ? :? "BE SURE SOURCE DISK IS IN DR
   IVE":? :? "BE SURE PRINTER IS READY":?
    :? :? "SELECT PAPER SIZE":?
DQ 710 ? "   (1) Standard  8 1/2 x 11 in
   ches."
MQ 720 ? :? "   (2) Legal size 8 1/2 x 14
    inches":POKE 752,1:? "?";
VV 730 GOSUB 660
UU 740 IF BYTE=49 THEN FS=66:NL=52:RETURN
YT 750 IF BYTE=50 THEN FS=84:NL=68:RETURN
PN 760 GOTO 730
TU 800 ? CHR$(125):? :? "WHICH DO YOU WAN
   T"
CK 810 ? :? "    (1) Change paper after ea
   ch page"
GL 820 ? :? "    (2) Continuous feed paper
   ":? :? "?"
VW 830 GOSUB 660
CV 840 IF BYTE=49 THEN CP=1:RETURN
ZM 850 IF BYTE=50 THEN CP=2:RETURN
PY 860 GOTO 830
```

Antic Classifieds

SOFTWARE, ETC.

★★★★★★

ENJOY quality PD software! Games—productivity—utilities—more! Demo disk, $3.00. . .catalog FREE. Homemade, 6011 Hyde Park Circle, Suite 206, Jacksonville, FL 32210 (6/89)

Atari 800/XL/XE public domain disks. High quality disks, fast service, free list. High quality disks, fast service, free list. 1st Byte, PO Box 130822, Tyler, TX 75713 (8/89)

Large library of the best public domain software for the 800/XL/XE. We have over 200 disks to pick from, low prices, fast service, & unbelievable discounts. Write for our large free list. ComputerActive, Box 893-A, Clinton, OK 73601 (6/89)

9 DISKS FULL OF SPARTADOS UTILITIES for use with 8 bit Atari & SpartaDOS. Only 14.95 + 2.50 S/H. ComputerActive, Box 893-S, Clinton, OK 73601 (6/89)

PRINT SHOP ICONS FOR THE ATARI 800/XL/XE. 10 disks with over 1000 pictures for use with Printshop. Only 14.95 + 2.50 S/H. Order today! ComputerActive, Box 893-P, Clinton, OK 73601 (6/89)

12 X-rated disks for the Atari 800/XL/XE only $14.95 + $2.50 S/H. You must send a signed statement of being over 21 with a check or money order to: ComputerActive, Box 893-X, Clinton, OK 73601 (6/89)

250+ DS/SD DISKS OF PD SOFTWARE FOR THE 8-BIT ATARI. SEND $3 TO: AAPDS 6-18TH EDGEMORE, HUTCHINSON, KS 67502 FOR A CATALOG LISTING AND DEMO DISK.

★★★★★★

BEST PD DISKS/PRICES—fastest service—FREE bonuses—8-bit/ST. Automatic DISK-A-MONTH program. Full disks less than $1.50 each. Super sample disk with catalog $3.50. Specify Computer. PATNOR, PO370782, El Paso, TX 79937(6/89)

The A Exchange—A Users Newsletter for ALL Atari Owners. Send SASE for FREE Issue. The A Exchange, P.O. Box 25852-A, Fresno, CA 93729 (6/89)

Public Domain Software—ST $4/Disk, 800/XL/XE $3/Disk. Largest ST Selection. The Atari ST Book $16.95. Fast Service, Visa/Mastercard, FREE catalog—Specify Computer. BRE Software, 352 W. Bedford, Suite 104-A, Fresno, CA 93711. (800) 347-6760 (1/90)

50 Arcade, Adventure, strategy games by Accolade, Atari, Avalon Hill, SSI, $6 each. Atari 800 48K and 810 drive $95. Paul Cirto (305) 920 1954 (6/89)

FREE ATARI 8-BIT PD SOFTWARE Send large SASE for listing and details you pay only for postage, handling and blank media. DINO'S RYO SOFTWARE, 664 W Arrow Hwy., Suite 197, San Dimas, Calif. 91773 (6/89)

★★★★★★

Stop overpaying for P.D. Software!!! ALL 8-bit disks only $2.00 each, ST only $3.00. Quantity discounts, great selection. Specify computer. Send large SASE: MWPDS, 890 N. Huntington St., Medina, OH 44256 (6/89)

NEW 800/XL/XE SOFTWARE! An international Collection of Shareware and Public Domain programs for the discriminating Atari owner. Every disk Guaranteed! Dependable worldwide service. Low prices! Send for Free descriptive catalogue. BELLCOM, Box 1043-A, Peterborough, Ontario, Canada K9J 7A5 (6/89)

STOP! LOOK! $4,000/month with computers. Homebased. Proven. No experience/capital. Guaranteed. Free details. MacKenzie, Box 91181—MCC, Pasadena CA 91109 (6/89)

AGAPE GAMES-5 CHRISTIAN GAMES. KINGS GLORY Arcade like Adventure with 26 mazes and a maze maker. BIBLE WORD SEARCHER with 30 puzzles and a puzzle generator. Plus 3 games that teach the Books of the Bible. For Atari 8-bit (requires 48K+disk) $12.95-Daniel Sharpe, GPO 911, Chicopee, MA 01021 (7/89)

COMPREHENSIVE COMIC BOOK SYSTEM! ATARI 8-bits, 48-K, disk drive. Excellent manual! $19.95. SASE for More Info. BUECO, 3900 Hampton Dr., Anchorage, AK 99504 (6/89)

★★★★★★

BATTERY BACKED UP RAMDISK. Never lose your memory, Boot from Ramdisk! No installation. Ramdisk cartridge plugs into Atari 400/800 or XL/XE. 32K $99.95, 64K $129.95, 192K $249.95 Certified check, Money order, VISA accepted. CANOE COMPUTER SERVICES, 11006-155 ST EDMONTON, AB, T5P-2N3 Phone (403) 437-4619 (6/89)

Your Ad Could Appear Here

Advertisers Index

Be sure to mention ANTIC when contacting these advertisers—all of whom support the Atari 8 bit and ST computer.

This list is provided as a courtesy to our advertisers. ANTIC does not guarantee accuracy or comprehensiveness.

Advertising Deadlines

August 1989

Insertion Orders: May 1
Artwork: May 8
On Sale: Last week of June

October 1989

Insertion Orders: June 30
Artwork: July 7
On Sale: Last week of August

September 1989

Insertion Orders: May 31
Artwork: June 7
On Sale: Last week of July

Tech Tips

GTIA PIXELS IN GR.0 & GR.8

BY GREG VOZZO

This short program shows you how to use those long, multi-colored GTIA pixels in other graphics modes.

There are many powerful uses for this technique, such as creating multi-colored redefined character sets for arcade games. The thickness of the GTIA pixels make letters and numbers almost illegible, but their variety of colors offer an easy way to add color to your characters.

In modes such as Graphics 8, each pixel is one bit wide. Thus, if a bit is set, its corresponding pixel is lit. In these graphics modes, the bit pattern shown in the left half of figure 1 will draw an "A" and uses 64 pixels.

GTIA pixels are four bits wide. In a GTIA mode, the bit pattern would draw a small multi-colored box eight scan lines tall and two pixels wide. The color of each pixel is determined by its bit pattern. The second scan line, for example, contains one pixel of COLOR 1 (or 0001 in binary) and one pixel of COLOR 8 (1000 in binary).

The right half of *Figure 1* shows the color breakdown of the entire character.

FIGURE I

DECIMAL	BITMAP	IN BINARY	AS SEEN BY GTIA	
0	--------	0000 0000	COLOR 0	COLOR 0
24	---XX---	0001 1000	COLOR 1	COLOR 8
60	--XXXX--	0011 1100	COLOR 3	COLOR 12
102	-XX--XX-	0110 0110	COLOR 6	COLOR 6
102	-XX--XX-	0110 0110	COLOR 6	COLOR 6
126	-XXXXXX-	0111 1110	COLOR 7	COLOR 14
102	-XX--XX-	0110 0110	COLOR 6	COLOR 6
0	--------	0000 0000	COLOR 0	COLOR 0

Learn these simple techniques and you'll have colorful GTIA characters dancing around your screen in no time.

The listing below is a short demo of using GTIA pixels in Graphics 0 and Graphics 8. When RUN, the program asks if you'd like to see mode 9 (16 shades of a single color), mode 10 (9 colors) or mode 11 (16 colors of a single shade). After you choose your mode, the program will draw a series of stripes in Graphics 8, print some text in the text window, then pop into your chosen GTIA mode and rotate the colors. When you're done, press any key to return to the main menu.

Don't type the TYPO II Codes!

```
RM 10 GRAPHICS 0:? "":? " 1 MODE 9":? :
   ? " 2 MODE 10":? :? " 3 MODE 11":? :
   ? :? "SELECTION:";
OD 20 CLOSE #1:OPEN #1,4,0,"K:"
VR 30 TRAP 30:GET #1,K:K=K-48:IF K<1 OR K
   >3 THEN 30
KB 40 G=K+8:GRAPHICS 8:POKE 87,G:FOR I=0
   TO 8+(G=15)*7
JE 50 COLOR I:FOR J=I*(10+(G=10)*5) TO I*
   (10+(G=10)*5)+(8+(G=10)*5)
SA 60 PLOT 0,J:DRAWTO 79,J:NEXT J:NEXT I:
   ? "THIS IS  TEXT  IN THIS MODE.";:POKE
    623,96*(K=1)+128*(K=2)+192*(K=3)
IN 70 IF K=1 THEN FOR I=0 TO 255:POKE 712
   ,I:GOSUB 75:NEXT I:POKE 712,0:GOTO 80
BV 71 IF K=2 THEN FOR I=704 TO 712:POKE I
   ,PEEK(53770):NEXT I:GOTO 80
YS 72 FOR I=0 TO 255:POKE 712,I:GOSUB 75:
   NEXT I:POKE 712,6:GOTO 80
KE 75 FOR P=1 TO 10:NEXT P:RETURN
SP 80 TRAP 80:GET #1,K:RUN
```

ATARI AUDIO CASSETTE

BY GREG VOZZO

This short BASIC program turns on your Atari 410 or 1010 Cassette Recorder's motor, letting you play any audio tape through the TV or monitor speaker. It reactivates itself whenever [RESET] is pressed. To deactivate it, POKE 1791 with any number from 0-254. To re-enable it, POKE 1791 with a 255.

If you want to disable it without disabling the [RESET] reactivation process, simply POKE a 60 into location 54018. POKE 54018 with a 52 to reactivate the motor at any time.

This program is an extension of my DOSINI HOUDINI, a Tech Tip from January 1988, which explains how to trap [RESET] by modifying part of your Atari's warm-start cycle.

Don't type the TYPO II Codes!

```
BR 10 FOR I=1775 TO 1791:READ A:POKE I,A:
   NEXT I:DATA 32,0,0,173,255,6,201,255,2
   08,5,169,52,141,2,211,96,255
YS 20 POKE 1776,PEEK(12):POKE 1777,PEEK(1
   3):POKE 12,239:POKE 13,6:? "CASSETTE M
   OTOR ENABLED":POKE 54018,52
```

Antic pays $25 for every original and exclusive Tech Tip submission that we publish. Send your 8-bit or ST disk and printout to: Antic Tech Tips, 544 Second Street, San Francisco, CA 94107. Tech Tips welcomes very short programs that demonstrate the Atari's powers, simple hardware modifications, or useful macros for popular software.

B&C ComputerVisions

3257 Kifer Road
Santa Clara, CA 95051
(408) 749-1003

STORE HOURS
TUE - FRI 10am - 6pm
SAT - 10am - 5pm
CLOSED SUN - MON

800/XL/XE SOFTWARE
ALL TITLES ON DISK

ENTERTAINMENT

12 ADAMS ADVENTURES .. 14.95
ALIANTS 26.95
ALT. REALITY CITY..... 26.95
ALT. REALITY DUNGEON. 26.95
BEYOND CASTLE WOLF.... 14.95
BISMARK 26.95
BOP & WRESTLE (64K)... 26.95
BORDINO:1812 .... 26.95
BOULDERDASH CONSTR.SET 17.95
BRUCE LEE 17.95
CASTLE WOLFENSTEIN.... 14.95
DALLAS QUEST 7.95
D-BUG 7.95
F-15 STRIKE EAGLE 31.50
FIGHT NIGHT 17.95
GAUNTLET (64K) 31.50
DEEPER DUNGEONS..... 22.50
GUNSLINGER 26.95
HARD HAT MAC 7.95
JAWBREAKER 9.95
KARATEKA 13.50
KNICKERBOCKERS 13.50
KORONIS RIFT 13.50
LAST V-8 .... 8.95
LEADERBOARD 13.50
MAIL ORDER MONSTERS .. 13.50
MICROLEAGUE BASEBALL.. 35.95
MONTEZUMA'S REVENGE... 14.95
MOUSEQUEST........... 17.95
MOON SHUTTLE 7.95
NINJA 8.95
OIL'S WELL 9.95
O'RILEY'S MINE 9.95
PIRATES OF BARB. COAST 22.50
PREPPIE I & II 9.95
RESCUE ON FRACTALAS... 13.50
SILENT SERVICE 31.50
SPEEDKING 8.95
SPIDERMAN 7.95
SPITFIRE 40 .... 31.50
STARFLEET I 44.95
SPY VS. SPY III 17.95
STOCKMARKET 22.50
STRIP POKER 26.95
SUMMER GAMES 17.95
TAX DODGE 9.95
THE HULK 7.95
TOMAHAWK (64K) 26.95
TOP GUNNER 17.95
TOUCHDOWN FOOTBALL ... 13.50
TRAILBLAZER 26.95
UNIVERSE 44.95
ZAXXON 13.50

PROGRAMMING

ACTION! 71.95
ACTION! TOOLKIT 26.95
BASIC XL 53.95
BASIC XL TOOLKIT 26.95
BASIC XE 71.95
DISK I/O 26.95
DRAPER PASCAL 44.95
KYAN PASCAL 62.95
LIGHTSPEED C 35.95
LOGO 29.95
MAC/65 .... 71.95
MAC/65 TOOLKIT 26.95
MACRO ASSEMBLER 22.50
PILOT 19.95
SPARTA DOS X 71.95
TOP DOS 1.5 PLUS 35.95

PRODUCTIVITY

ANIMATION STATION 89.95
ATARIWRITER+. 39.95
ATARI BOOKKEEPER 24.95
ATARI MUSIC II 14.95
AWARDWARE (1050) 13.50
BANK STREET WRITER.... 14.95
BLAZING PADDLES 31.50
CELEBRITY COOKBOOK ... 26.95
COMPUTE YOUR ROOTS ... 35.95
DATAMANAGER 17.95
FAMILY FINANCE 6.95
GUITAR WIZARD 26.95
HOME ACCOUNTANT 19.95
HOME FILING MANAGER. . 6.95
HOMEPAK 24.95
INVENTORY MASTER 80.95
LETTER WIZARD 29.95
MUSIC CONSTRUCTION SET 13.50
NEWSROOM (1050 - 64K). 13.50
NEWS STATION 26.95
NEWS STA. COMPANION. . 26.95
PAGE DESIGNER 26.95
PRINT POWER (1050).... 13.50
PRINTKIT (1050) 13.50
PRINTSHOP 34.95
P.S. COMPANION (64K) . 24.95
P.S.GRAPHICS LIBRARY 1 17.95
P.S.GRAPHICS LIBRARY 2 17.95
P.S.GRAPHICS LIBRARY 3 17.95
PROOF READER 17.95
PUBLISHING PRO 35.95
RUBBER STAMP 26.95
SYNTREND 14.95
SUPER MAILER 35.95
THE LOTTO PROGRAM 17.95
TIMEWISE 6.95
TURBOWORD/80 COLUMN
REQUIRES XEP80....... 44.95
VIDEO TITLESHOP (64K). 26.95
GRAPHICS COMPANION. 17.95
VIRTUOSO 29.95
VISICALC 24.95

EDUCATION

ATARI LIGHT MODULE
(REQ.ATARILAB STARTER) 9.95
BUZZWORD 35.95
GRANDMA'S HOUSE (-10) 9.95
HEY DIDDLE (AGE 3-10). 9.95
MASTER TYPE 14.95
PLANATARIUM 22.50
STATES AND CAPITALS .. 9.95
TOUCH TYPING 9.95

CBS (AGE 3-6):
ASTROGROVER 8.95
BIG BIRD SPEC DELIVE 8.95
ERNIE'S MAGIC SHAPE. 8.95

DESIGNWARE:
MATHMAZE (6-11) 35.95
MISSION ALGEBRA (13+)35.95
SPELLICOPTER (6-11). 35.95

TINK TONK (AGE 4-6):
ABC'S 8.95
COUNT AND ADD 8.95
SMART THINKER 8.95
SPELLING 8.95
SUBTRACTION 8.95
THINKING SKILLS 8.95
ALL 6 TINK TONKS.. 39.95

UNICORN:
10 LITTLE ROBOTS
(PRE-SCHOOL) 26.95
FUN BUNCH (6-ADULT) 26.95
RACECAR RITHMETIC
(AGE 6+) 26.95

WEEKLY READER (PRE-SCHOOL):
STICKY BEAR SHAPES . 26.95
STICKY BEAR NUMBERS. 26.95
STICKY BEAR ABC'S .. 26.96
STICKY BEAR OPPOSITE 26.95

800/XL/XE SOFTWARE
ALL TITLES ON CARTRIDGE

ENTERTAINMENT

ALIEN AMBUSH 9.95
ACE OF ACES .(XL/XE) . 24.95
ARCHON 19.95
ASTEROIDS 15.95
ATARI TENNIS 9.95
BALL BLAZER 19.95
BARNYARD BLASTER 24.95*
BATTLEZONE 19.95
BLUE MAX 19.95
CAVERNS OF MARS 14.95
CENTIPEDE 14.95
CHICKEN 9.95
CHOPLIFTER 14.95
CLAIM JUMPER 9.95
CLOUDBURST 9.95
CRIME BUSTER(XL/XE) . 24.95*
CROSSBOW 24.95*
CROSSFIRE 9.95
DAVIDS MIDNIGHT MAGIC 19.95
DEFENDER 14.95
DELUXE INVADERS 7.95
DESERT FALCON 19.95
DIG DUG 19.95
DONKEY KONG 5.00
DONKEY KONG JR. 19.95
EASTERN FRONT (1941). 19.95
E.T. PHONE HOME 9.95
FIGHT NIGHT 19.95
FINAL LEGACY 19.95
FOOD FIGHT (XL/XE)... 19.95
FOOTBALL 14.95
FROGGER 14.95
GALAXIAN 19.95
GATO 24.95
GORF (400/800) 5.00
GYRUSS 14.95
HARDBALL 19.95
INTO THE EAGLES NEST 19.95
JOURNEY TO PLANETS .. 9.95
JOUST 19.95
JUNGLE HUNT 19.95
KABOOM! 14.95
LODE RUNNER 24.95
MARIO BROS. 19.95
MILLIPEDE 19.95
MISSILE COMMAND 5.00
MOON PATROL 19.95
MR. COOL 9.95
MS. PAC MAN 19.95
NECROMANCER 19.95
ONE ON ONE (XL/XE)... 19.95
PAC MAN 5.00
PENGO 19.95
POLE POSITION 19.95
POPEYE 14.95
Q-BERT 14.95
QIX 14.95
RESCUE ON FRACTALAS . 19.95
RETURN OF THE JEDI .. 14.95
ROBOTRON:2084 .... 19.95
SKY WRITER 14.95
SLIME (400/800)...... 9.95
SPACE INVADERS 14.95
STAR RAIDERS 5.00
STAR RAIDERS II 19.95
SUPER BREAKOUT 9.95
TRACK & FIELD 24.95
TURMOIL 9.95
WIZARD OF WOR........ 5.00
* REQUIRES LIGHT GUN

PRODUCTIVITY

ATARIWRITER 19.95
MICROFILER 22.50

EDUCATION

ATARILAB STARTEER SET 29.95
MATH ENCOUNTERS 9.95

FISHER PRICE (PRE SCHOOL):
DANCE FANTASY 8.95
LINKING LOGIC 8.95
LOGIC LEVELS 8.95
MEMORY MANOR 8.95

SPINNAKER (AGE 3-10):
ALF IN COLOR CAVES . 9.95
ALPHABET ZOO 9.95
DELTA DRAWING 9.95
FACEMAKER 9.95
KIDS ON KEYS 9.95
KINDERCOMP 9.95

(AGE 7 - ADULT):
ADV.CREATOR (400/800).9.95
FRACTION FEVER 9.95

THE BASIC TUTOR
Learn to program in BASIC
Requires a 410 OR 1010 Program Recorder
COMPLETE PACKAGE ONLY **$9.95**

BOOKS ONLY

DE RE ATARI 10.00
LOGO 10.00
ATARIWRITER 10.00
DOS 2.5 .... 12.95
BASIC REFERENCE 5.00
BOOKKEEPER 10.00

SUPER SPECIALS

RECONDITIONED ATARI MERCHANDISE
30 DAY WARRANTY

800 (48K) COMPUTER $79.95	**SPACE AGE JOYSTICK $5.00**	**1030 MODEM WITH EXPRESS! $24.95**
400 (16K) COMPUTER $29.95	**ATARI TRACKBALL $9.95**	**1010 PROGRAM RECORDER $29.95**
1020 COLOR PRINTER/PLOTTER **$19.95** 40 COLUMNS WIDE (NEW IN BOX)	**ATARI BOOKKEEPER $14.95 - NO BOX**	**DISKETTES AS LOW AS 20 CENTS** 10 FOR $4.00 100 FOR $29.95 1000 FOR $200 MOST ARE UNNOTCHED WITH OLD SOFTWARE
	ATARI NUMERIC KEYPAD $7.95	

SHIPPING INFORMATION - Prices do not include shipping and handling. Add $5.00 for small items ($8.00 Min. for Canada). Add $8.00 for disk drive. Add $2.75 for C.O.D. Calif. res. include 7% sales tax. Mastercard and Visa accepted if your telephone is listed in your local phone directory. Orders may be pre-paid with money order, cashier check, or personal check. Personal checks are held for three weeks before order is processed. C.O.D orders are shipped via UPS and must be paid with cash, cashier check or money order. International and APO orders must be pre-paid with cashier check or money order. $20.00 minimum on all orders. All sales are final - no refunds - prices are subject to change.

Phone orders accepted **TUESDAY THROUGH FRIDAY** from 10:00 am to 6:00 pm PST.

We carry a complete line of ATARI products and have a large public domain library.
Write or call for free catalogue. (408) 749-1003 TUE - FRI 10AM - 6 PM

PRICES SUBJECT TO CHANGE WITHOUT NOTICE - ALL SALES ARE FINAL

BARGAINS!

1020 COLOR PRINTER PLOTTER
Complete with:
- 2 Pen Sets
- 1 Roll Paper
- Power Supply & Cable

$14.98 Brand New

EXTRA PEN SETS .... $ .89 (Black) $3.98 (Color)

1200XL 64K COMPUTER w/PAC-MAN **$49.** Reconditioned

1027 PRINTER
- 80 Column
- Direct Connect
- Letter Quality

$79. Brand New

SYNTREND $4.98

850 Interface Serial, parallel **$79.98** Reconditioned

825 80 Column Printer
- Friction, Trac
- Dot Matrix

$69. Reconditioned

BASIC TUTOR 4 BOOKS **$4.98**

ARCADE CHAMP
- 2 JOYSTICKS
- QIX
- PAC-MAN

$14.98

NEW **DOS XE** .. **$9.95**

1010 Tape drive .... $29.
10' Joystick ext. .... $ .99
Trackball $9.95

ATTN. DEALERS

PAC-MAN Case of 120 **$69.**

1050 DISK DRIVE **$169.** Reconditioned

800 48K COMPUTER w/PAC-MAN **$69.95** Reconditioned

1025 PRINTER
- 80 Column
- Dot Matrix
- Friction/Tractor
- Direct Connect

$79.98 Reconditioned

800 BOARDS
- Mother
- CPU
- Power
- ROM

YOUR CHOICE **$8.98**

CARTRIDGES FOR 800, XL, XE

BASIC CARTRIDGE	$4.95	DELTA DRAWING	$8.98	BATTLEZONE	$19.98
BASIC TUTOR (4 BOOKS)	$4.95	HEY DIDDLE DIDDLE	$8.98	FOOD FIGHT	$19.98
TURMOIL	$4.95	SLIME (400/800)	$8.98	HARDBALL	$19.98
PAC-MAN (no box)	$4.95	ALPHABET ZOO	$8.98	FIGHT NIGHT	$19.98
DONKEY KONG (no box)	$4.95	ALF	$8.98	ONE ON ONE BASKETBALL	$19.98
GORF (400.800)	$4.95	ADVENTURE CREATOR	$8.98	DESERT FALCON	$19.98
DEMON ATTACK (400.800)	$4.95	SKY WRITER	$14.95	NECROMANCER	$19.98
DELUXE INVADERS	$4.95	FOOTBALL	$14.95	RESCUE ON FRACTALUS	$19.98
JOURNEY TO THE PLANETS	$4.95	DEFENDER	$14.95	BALLBLAZER	$19.98
MATH ENCOUNTER	$7.98	ROBOTRON	$19.98	BLUE MAX	$19.98
DANCE FANTASY	$8.98	TENNIS	$19.98	STAR RAIDERS II	$19.98
LOGIC LEVELS	$8.98	FINAL LEGACY	$19.98	DAVID'S MIDNIGHT MAGIC	$19.98
MEMORY MANOR	$8.98	MARIO BROS.	$19.98	ARCHON	$19.98
LINKING LOGIC	$8.98	DONKEY KONG JR.	$19.98	GATO	$24.98
CHICKEN	$8.98	JUNGLE HUNT	$19.98	ACE OF ACES	$24.98
CLAIM JUMPER	$8.98	MOON PATROL	$19.98	LODE RUNNER	$24.98
				BARNYARD BLASTER (XE gun)	$24.98

DISK SOFTWARE FOR 800, XL, XE

DAVID'S MIDNIGHT MAGIC	$4.98	CROSSCHECK	$4.98	SPIDERMAN	$4.98
ZORRO	$4.98	MOLECULE MAN	$4.98	HULK	$4.98
BANDITS (48K 400.800)	$4.98	CRYSTAL RAIDER	$4.98	VISICALC	$24.98
PROTECTOR II	$4.98	DESPATCH RIDER	$4.98	BOOKKEEPER W/ numeric keypad	$29.98
CLAIM JUMPER	$4.98	MISSION ASTEROID	$4.98	GET RICH	$39.98
SYNTREND	$4.98				

810 DISK DRIVES **$129.**

8-BIT I/O CABLE $4.98

SAN JOSE COMPUTER
T H E A T A R I S T O R E
Sunrise Plaza 640 Blossom Hill Rd. San Jose, CA 95123
(408) 224-8575 • BBS (408) 224-9052

5.25" DISKS
20 CENTS EA.*

QTY.	PRICE
10	$4.00
100	$29.95
*1000	$200

MAJORITY ARE UNNOTCHED CONTAINING OLD SOFTWARE

SHIPPING: ADD $5.00 TO ALL ORDERS. AIR AND INTERNATIONAL SHIPPING EXTRA. THAT'S IT.
WARRANTY: 90 DAY WARRANTY ON ALL ITEMS. **TAX:** CALIFORNIA RESIDENTS ADD 7% SALES TAX.
PREPAYMENT: USE VISA, MASTERCARD, MONEY ORDER, CASHIER'S CHECK OR PERSONAL CHECK.
PERSONAL CHECK MUST CLEAR PRIOR TO SHIPMENT. **C.O.D.:** CASH, CASHIER'S CHECK OR M.O. ONLY.
Prices subject to change without notice. Brand and/or product names are trademarks or registered trademarks of their respective holders.
Ad produced on an ATARI ST.

When You Want To Talk Atari... You Want To Talk To Us.

COMPUTERS

Atari
65XE $109
130XE 149

INTERFACES

ICD
P:R Connection 58
Printer Connection 34
850 To Par Cable 15
Supra
1150 36
Xetec
Graphix Interface 38
Atari
850 Interface 109

XL/XE ENHANCEMENTS

Axlon
32K Memory 400/800 16
Atari
80 Column Card 69
I/O Cable 7
ICD
Sparta DOS Construction Set 26
US Doubler/Sparta DOS 45
Real Time Clock 46
Rambo XL 26

MODEMS

Atari
SX212 300/1200 ST Dir. Con. 64
XM301 44
Avatex
1200HC 89
2400 159
Supra
2400 Baud XL/XE or ST 159
2400 Baud (No Software) 139

DRIVES

Atari
XF551 XL/XE 179
I.B.
5¼ 40 Track (ST) 219
I.C.D.
FA-ST 20 Meg 629
FA-ST 30 Meg 759
Indus
GTS 100 3½" DS/DD (ST) 189
GT 1000 5¼" DS/DD (ST) 199
Supra
30 Meg Hard Drive (ST) 649
60 Meg Hard Drive (ST) 989

PRINTERS

Alps
ASP 1000 120cps, 9-Pin 179
Atari
XDM121 Letter Quality XL/XE $189
Brother
M-1109 100cps Dot Matrix 159
Epson
LX-800 150cps, 80 Col. 189
NEC
P2200 Pinwriter 24-Wire 349
Okidata
ML-390+ 270cps 24-Wire 509
Panasonic
KX-P1091i 194cps, 80 Col. 179
Star Micronics
NX-1000 140cps, 80 Col. 169
NX-1000 Rainbow Color 239

SOFTWARE XL/XE

Access
Leaderboard Golf 13
Atari
Atariwriter Plus 36
Broderbund
The Print Shop 27
Microprose
Top Gunner 16
F-15 Strike Eagle 23

SOFTWARE (ST)

Accolade
Test Drive 24
Activision
Hacker 2 Doomsday 29
Antic
Flash 19
Avant Garde
PC Ditto 59
Cygnus
Starfleet 34
DataEast
Speed Buggy 27
Electronic Arts
Auto Duel 32
Epyx
Divebomber 19
Firebird
Tracker 14
FTL
Dungeonmaster 27
Michtron
Leatherneck 28
Microprose
F-15 Strike Eagle 24
Psygnosis
Obliterator 27
Soft Logik Corp.
Page Stream 139
Timeworks
Desktop Publisher 79
WordWriter 45
Unison World
Printmaster Plus 23

EXCLUSIVE XL/XE SOFTWARE FINAL CLOSEOUT



ATARI 1020 Printer, 40 Col. Color $19

"Purchase $20.00 of the following items & pay only $10.00 for the Atari 1020 Color Printer"

AA4204 1010 Color Pens $2.49
AA4206 1020 Color Pens 2.49

UNBOXED/UNDOCUMENTED CARTRIDGES

AAB4008 Space Invaders 1.99
AAB4011 Star Raiders 1.99
AAB4012 Missile Comnd. 1.99
AAB4013 Asteroids 1.99
AAB4022 Pac Man 1.99
AAB4024 Galaxian $1.99
AAB4025 Qix 1.99
AAB8030 E.T. Phone Ho. 1.99
AAB8039 Eastern Front 1.99

DOCUMENTED SOFTWARE

AA4011 Star Raiders 2.99
AA4012 Missile Command 2.99
AA4022 Pac Man 4.99
AA4025 Defender 4.99
AA4102 Kingdom (C) 1.99
AA4126 Speed Reading 2.49
AA4112 States & Cap. (C) 1.99
AA4129 Vuggles Rainbow 2.49
AA415 Filemanager 6.99
AA5047 Timewise (D) 2.99
AA6006 Counseling Proc. 1.99
AA8048 Millipede 4.99
AA9035 Mickey Outdoors (C) 3.99

ATARI PROGRAM EXCHANGE (CASSETTES)

Counter - Downhill - Dog Daze - Quaxon - Salmon Run - Typo Attack - Video Math Flashcard - Fig/Forth 1.49/ea.

ROKLYN

Anti-Sub (D) & Journey to Planet (R) 3.49/ea.

Call today on miscellaneous Mindscape, Data Soft, and Synapse Titles as low as $1.99/ea.

CMO

Your Source for Hardware, Software & Peripherals

In U.S.A.

800-233-8950

In Canada call: 800-233-8949
All Other Areas call: 717-327-9575 Fax call: 717-327-1217
Educational, Governmental and Corporate Organizations
Call toll-free: 1-800-221-4283
CMO, 101 Reighard Ave., Dept. B6, Williamsport, PA 17701

Over 400,000 SATISFIED CUSTOMERS • MAJOR CREDIT CARDS ACCEPTED

MEMBER MMC MICROCOMPUTER MARKETING COUNCIL of the Direct Marketing Association Inc

POLICY: Add 3% (minimum $7.00) shipping and handling. Larger shipments may require additional charges. Personal and company checks require 3 weeks to clear. For faster delivery, use your credit card or send cashier's check or bank money order. Credit cards are not charged until we ship. Pennsylvania and Maryland residents add appropriate sales tax. All prices are U.S.A. prices and are subject to change. All items are subject to availability. Defective software will be replaced with the same item only. Hardware will be replaced or repaired at our discretion within the terms and limits of the manufacturer's warranty. We cannot guarantee compatibility. All sales are final and returned shipments are subject to a restocking fee. We are not responsible for typographic or photographic errors.

B606